Computer Networks

Quanlong Li

Chapter 1: introduction

our goal:

- get "feel" and terminology
- more depth, detail *later* in course
- approach:
  - use Internet as example

overview:

- what's the Internet?
- what's a protocol?
- network edge; hosts, access net, physical media
- network core: packet/circuit switching, Internet structure
- performance: loss, delay, throughput
- protocol layers, service models
- history

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

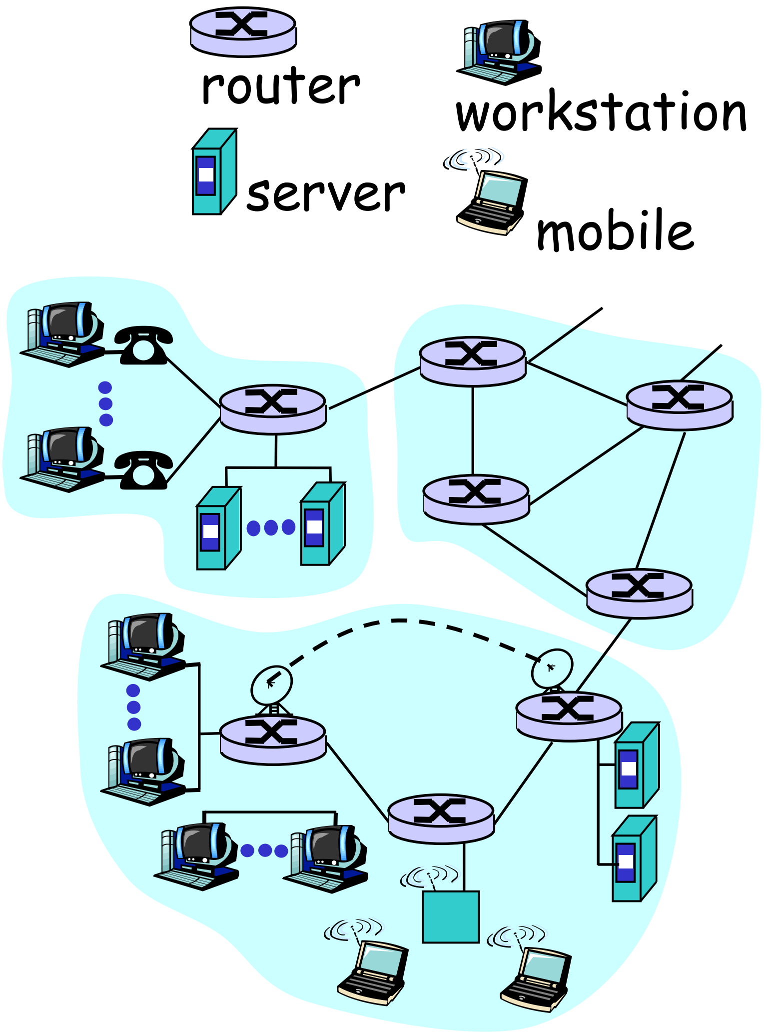
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 history

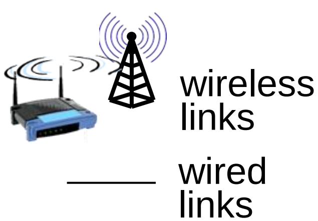
What's a Computer Network?

- Sets of Connected Autonomous computing devices(computers)
- Connected –*must be connected by communication links(such as physical media)*
- Autonomous-*each hosts works independently (not controlled by other ones)*

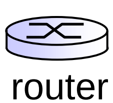


What's the Internet: "nuts and bolts" view

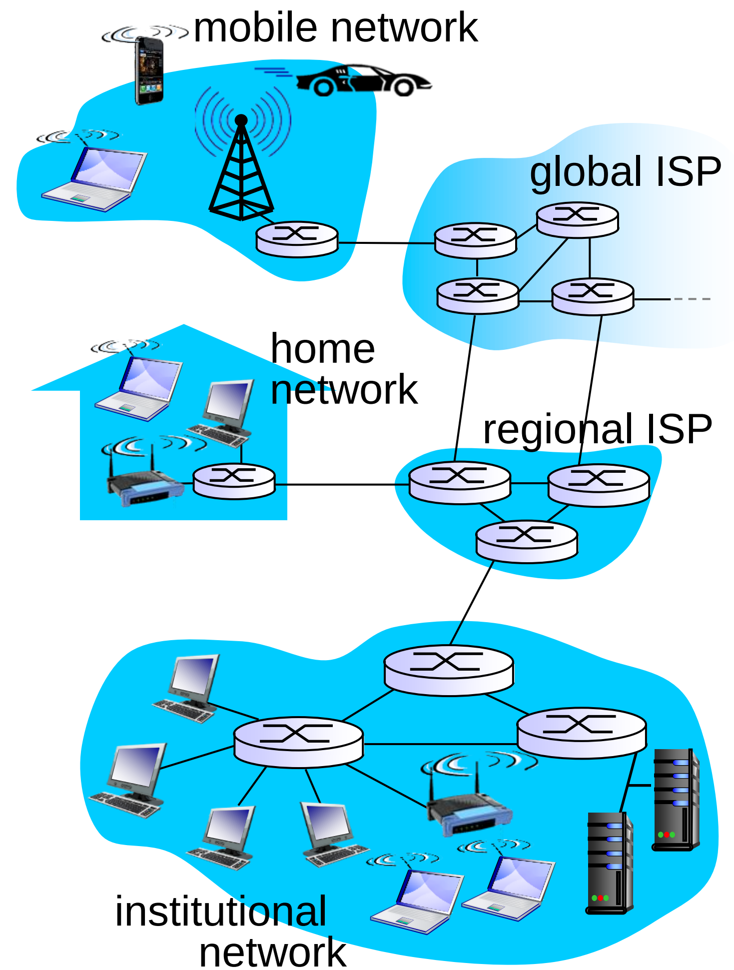
- millions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*



- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate: *bandwidth*



- *Packet switches:* forward packets (chunks of data)
  - *routers* and *switches*



“Fun” internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster +
weather forecaster

Tweet-a-watt:
monitor energy use

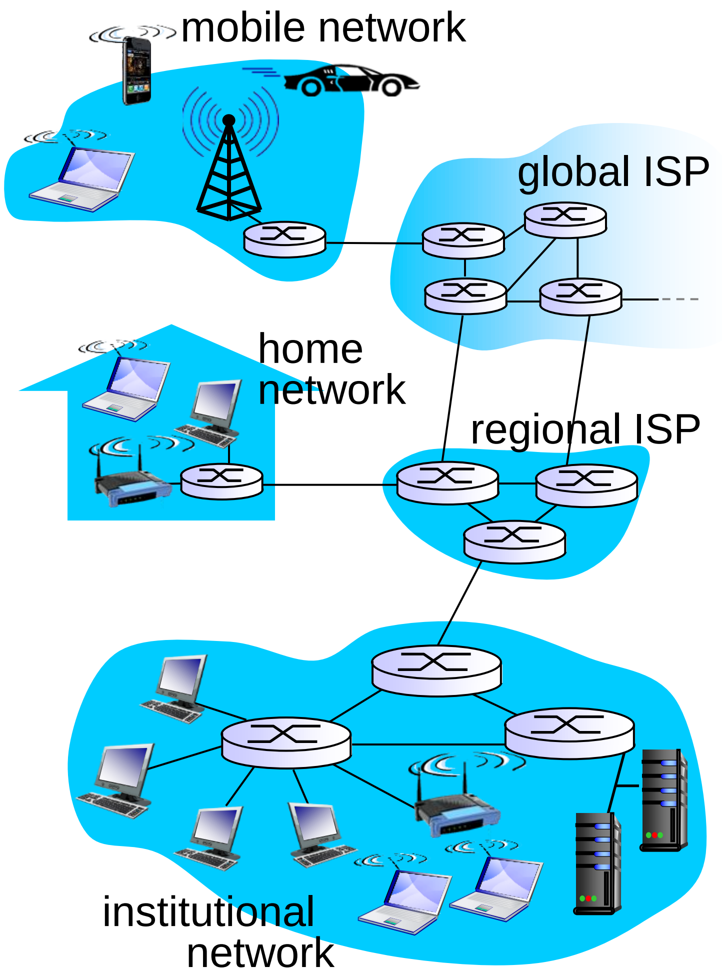
Internet
refrigerator

Slingbox: watch,
control cable TV remotely

Internet phones

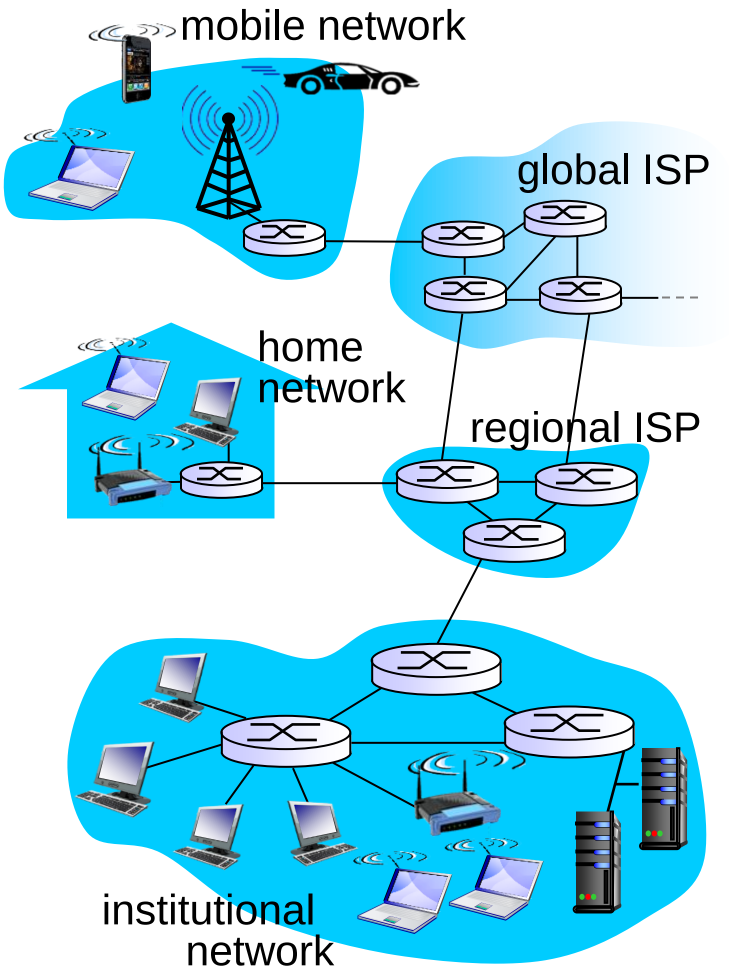
What’s the Internet: “nuts and bolts” view

- *Internet: “network of networks”*
  - Interconnected ISPs
- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, 802.11
- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet: a service view

- *Infrastructure that provides services to applications:*
  - Web, VoIP, email, games, e-commerce, social nets, …
- *provides programming interface to apps*
  - hooks that allow sending and receiving app programs to "connect" to Internet
  - provides service options, analogous to postal service



What's a protocol?

human protocols:

- "what's the time?"
- "I have a question"
- introductions

… specific msgs sent

… specific actions taken when msgs received, or other events

network protocols:

- machines rather than humans
- all communication activity in Internet governed by protocols

protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

Key Elements of a Protocol

- Syntax
  - Data formats
  - Signal levels
- Semantics
  - Control information
  - Error handling
- Timing
  - Speed matching
  - Sequencing

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

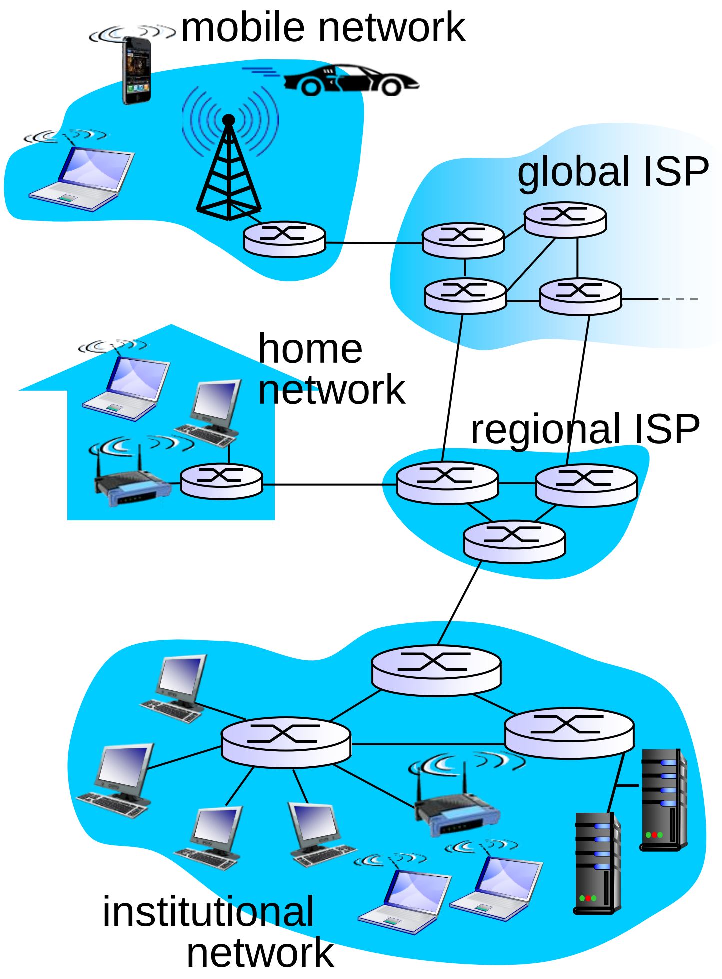
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 history

A closer look at network structure:

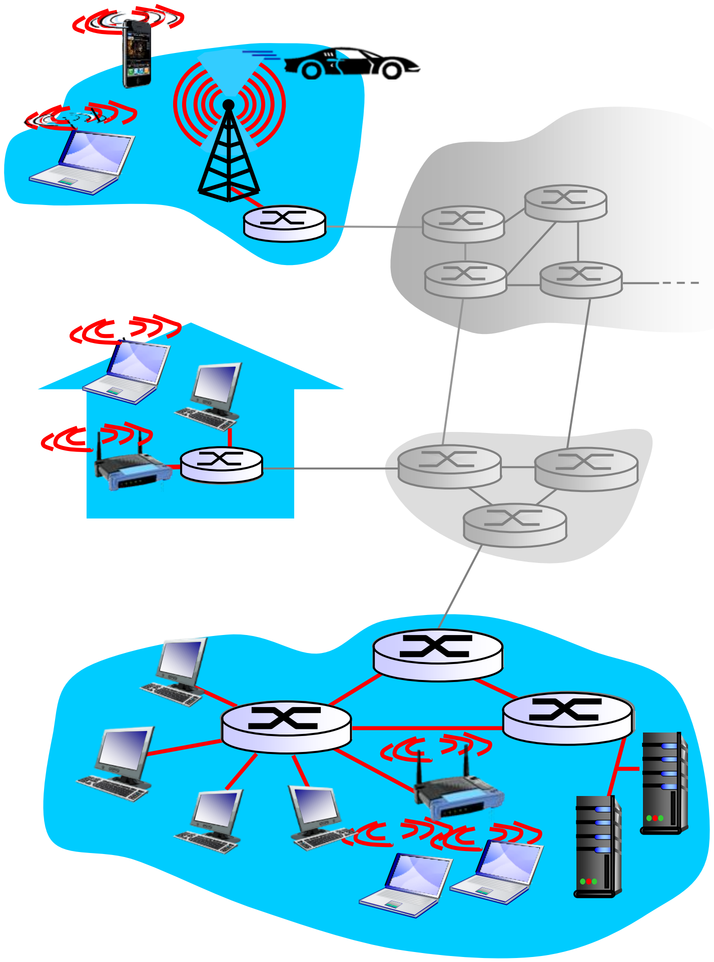
- *network edge:*
  - hosts: clients and servers
  - servers often in data centers
- *access networks, physical media:* wired, wireless communication links
- *network core:*
  - interconnected routers
  - network of networks



Access networks

Q: How to connect end systems to edge router?

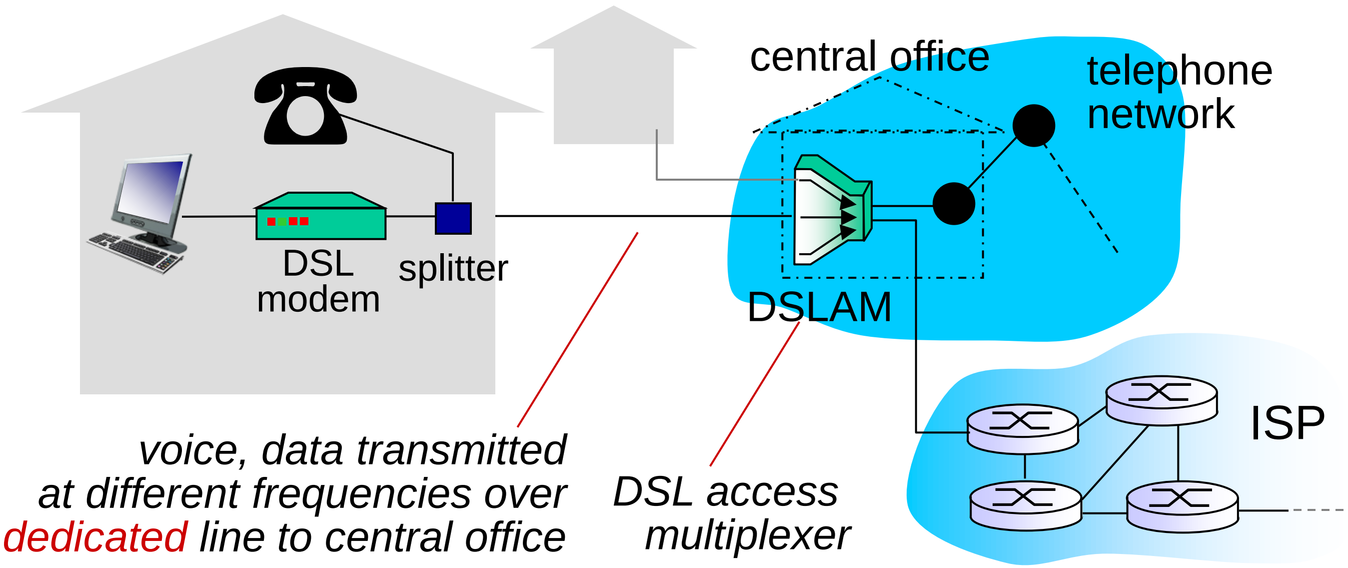
- residential access nets
- institutional access networks (school, company)
- mobile access networks



keep in mind:

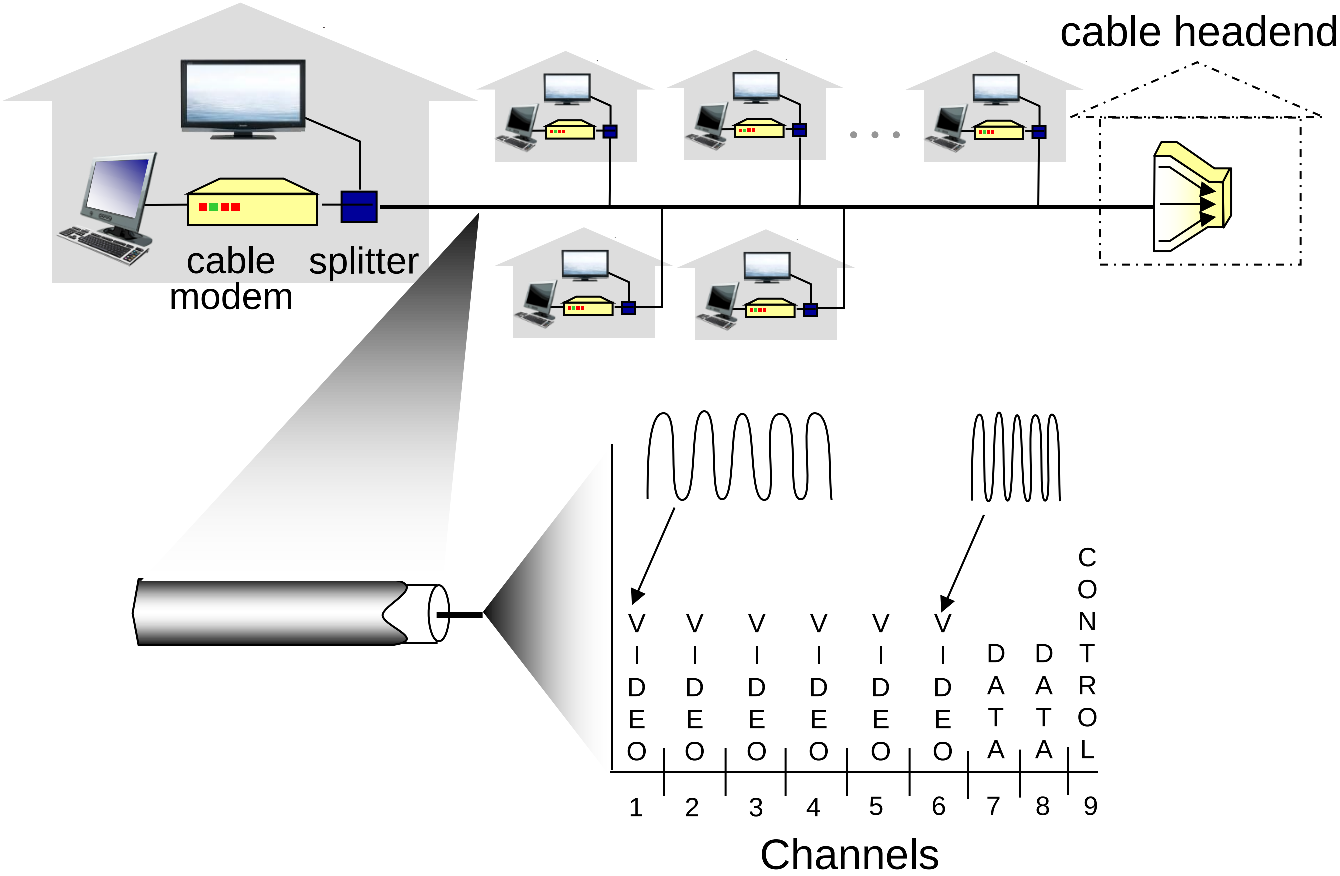
- bandwidth (bits per second) of access network?
- shared or dedicated?

Access net: digital subscriber line (DSL)



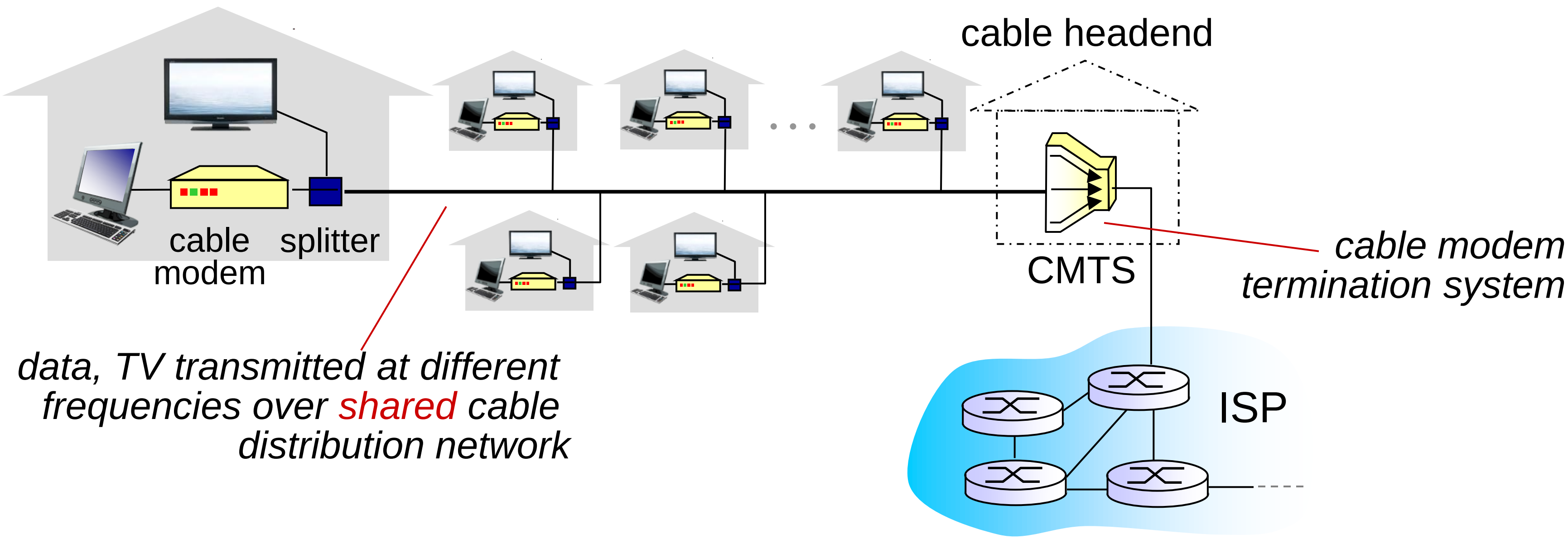
- use *existing* telephone line to central office DSLAM
  - data over DSL phone line goes to Internet
  - voice over DSL phone line goes to telephone net
- < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
- < 24 Mbps downstream transmission rate (typically < 10 Mbps)
- FDM: >50 kHz - 1 MHz for downstream
  4 kHz - 50 kHz for upstream
  0 kHz - 4 kHz for ordinary telephone

Access net: cable network



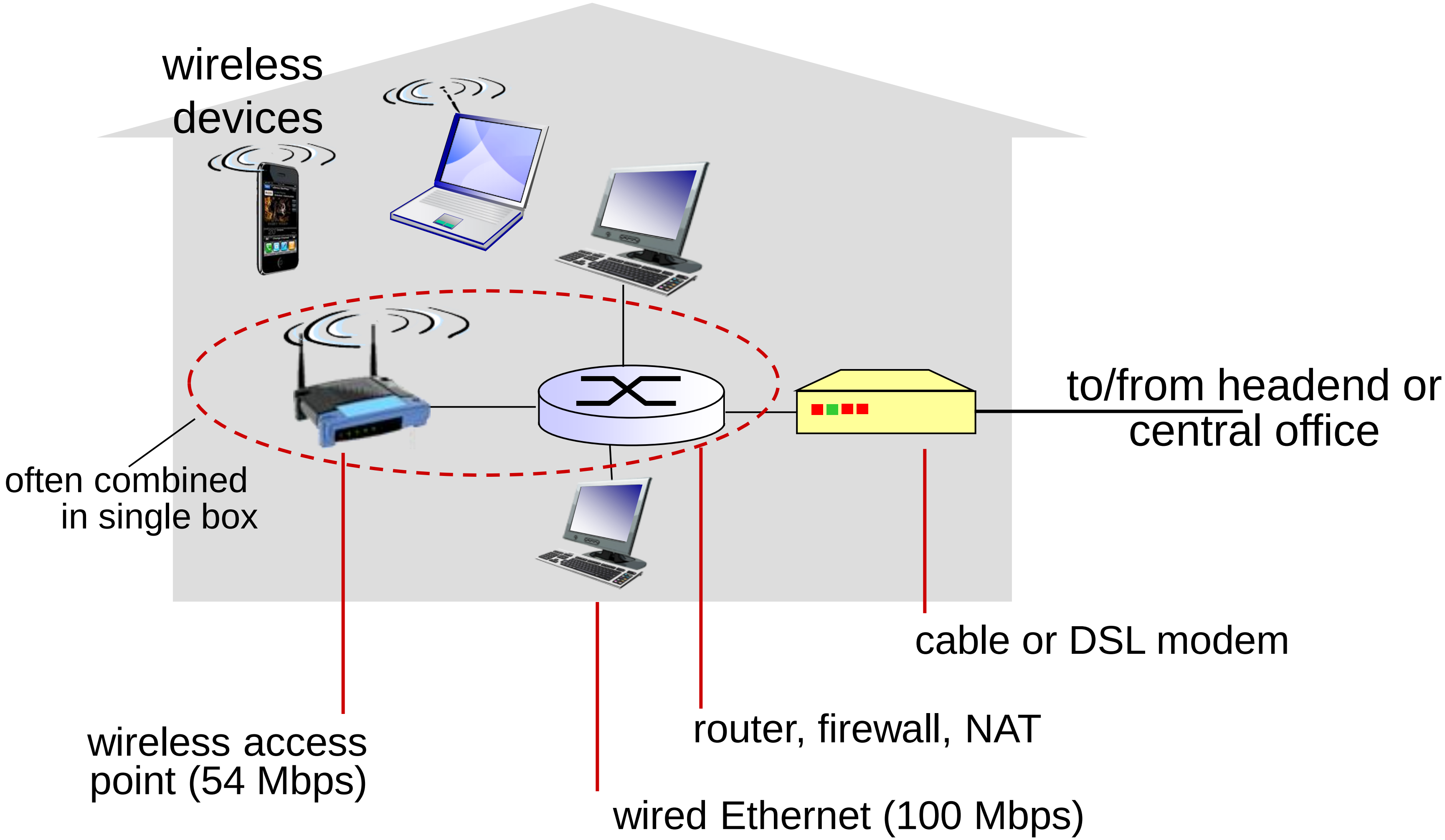
frequency division multiplexing: different channels transmitted in different frequency bands

Access net: cable network

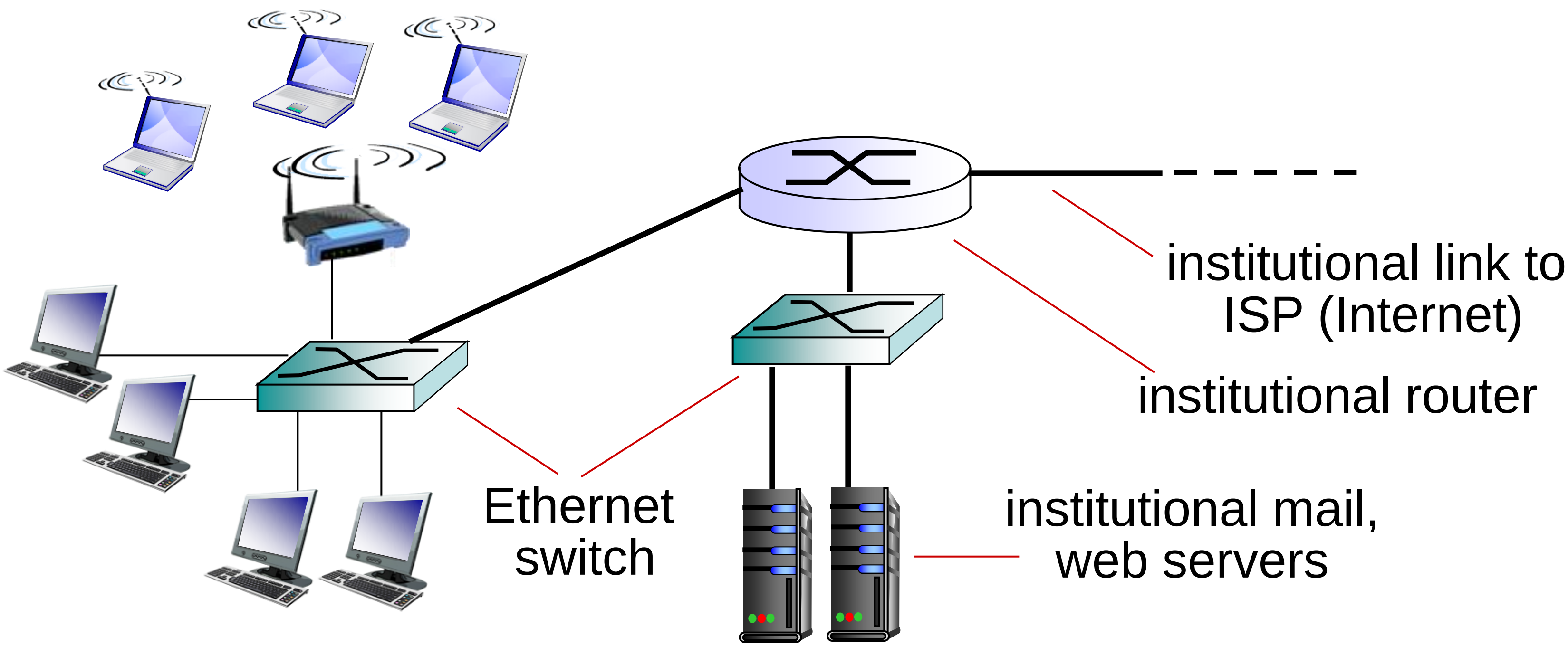


- HFC: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream transmission rate, 2 Mbps upstream transmission rate
- network of cable, fiber attaches homes to ISP router
  - homes *share access network* to cable headend
  - unlike DSL, which has dedicated access to central office

Access net: home network



Enterprise access networks (Ethernet)

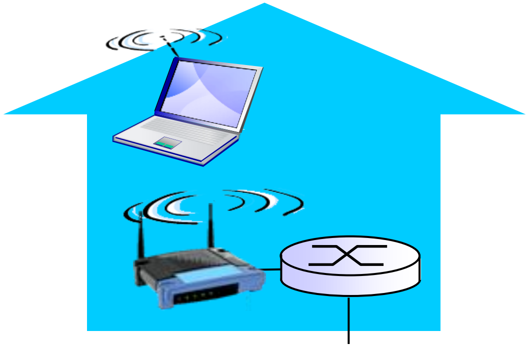


- typically used in companies, universities, etc
- 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- today, end systems typically connect into Ethernet switch

Wireless access networks

- shared *wireless* access network connects end system to router
  - via base station aka "access point"

wireless LANs:
- within building (100 ft)
- 802.11b/g (WiFi): 11, 54 Mbps transmission rate



wide-area wireless access
- provided by telco (cellular) operator, 10's km
- between 1 and 10 Mbps
- 3G, 4G: LTE

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

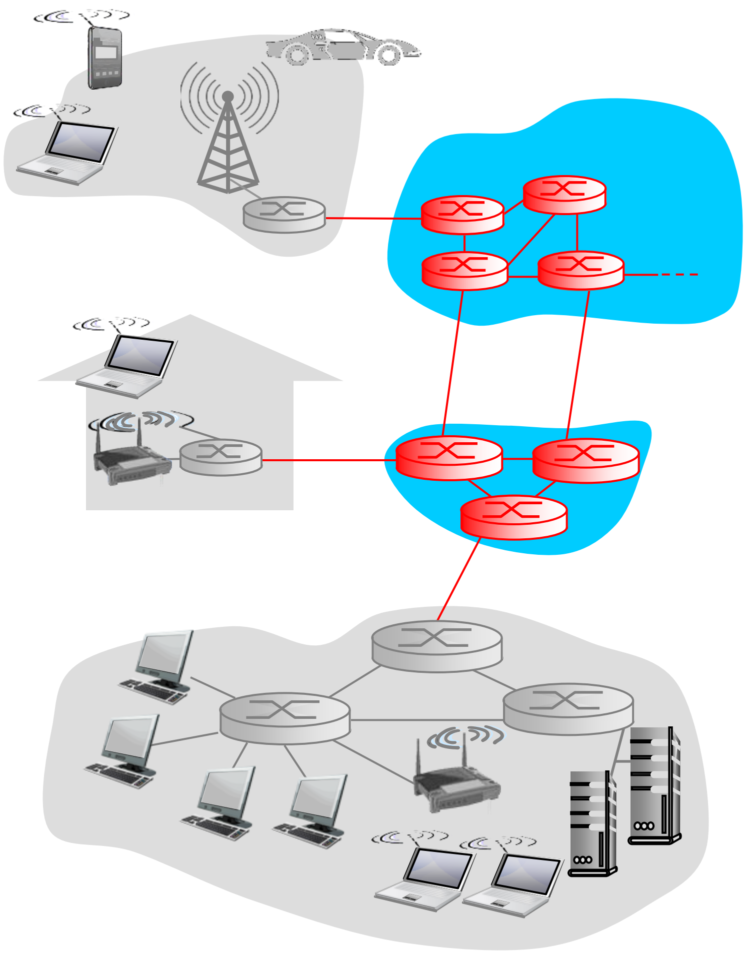
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 history

The network core

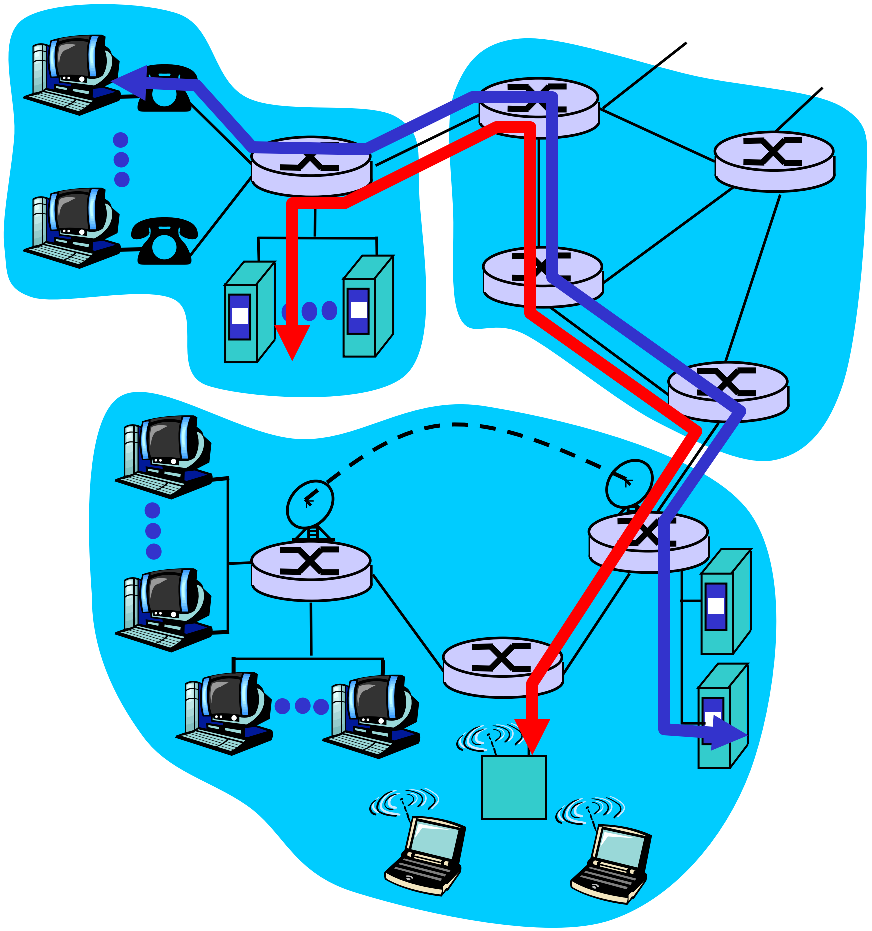
- mesh of interconnected routers
- the fundamental question: how is data transferred through net?
  - Switching!
    - Circuit switching
    - Message switching
    - Packet switching



Network Core: Circuit Switching

- Call setup→ communication→ tear down
- dedicated resources: no sharing
  - link bandwidth, switch capacity
- circuit-like (guaranteed) performance
- call setup required

End-end resources reserved for "call"



Network Core: Circuit Switching

How to share the trunk line?

--Multiplexing: network resources (e.g., bandwidth) divided into "pieces"

- pieces allocated to calls
- resource piece *idle* if not used by owning call *(no sharing)*

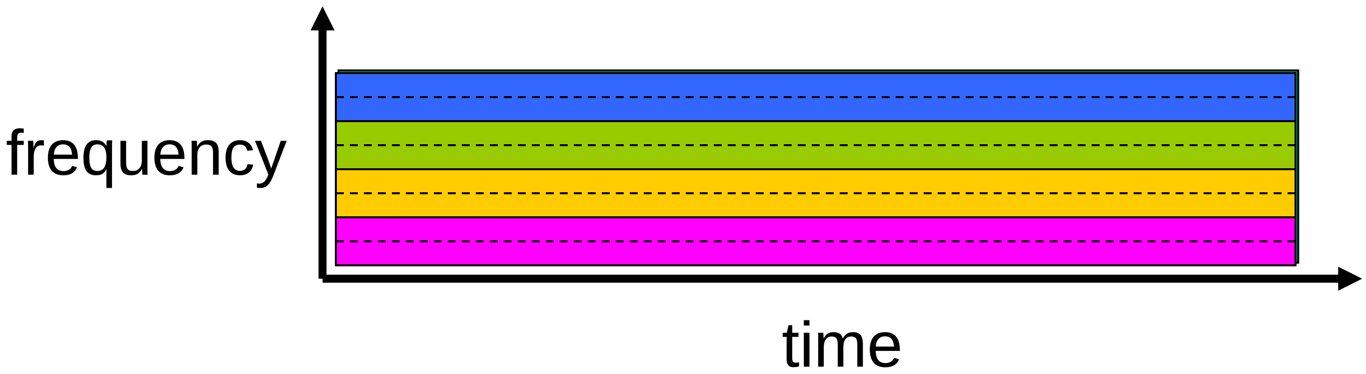
- Typical multiplexing methods:
  - frequency division multiplexing-FDM
  - time division multiplexing-TDM
  - Wavelength division multiplexing-WDM
  - Code division multiplexing-CDM

Circuit switching: FDM versus TDM

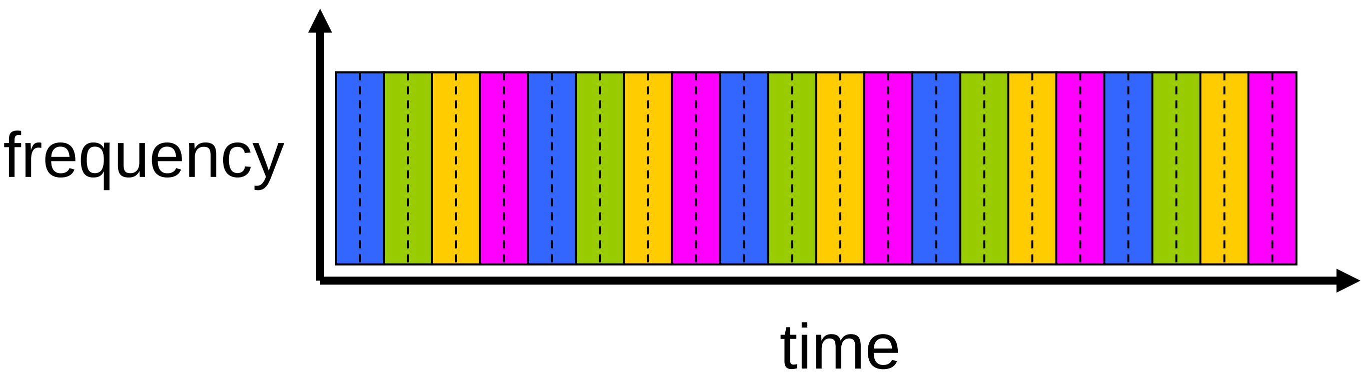
Example:

4 users

FDM

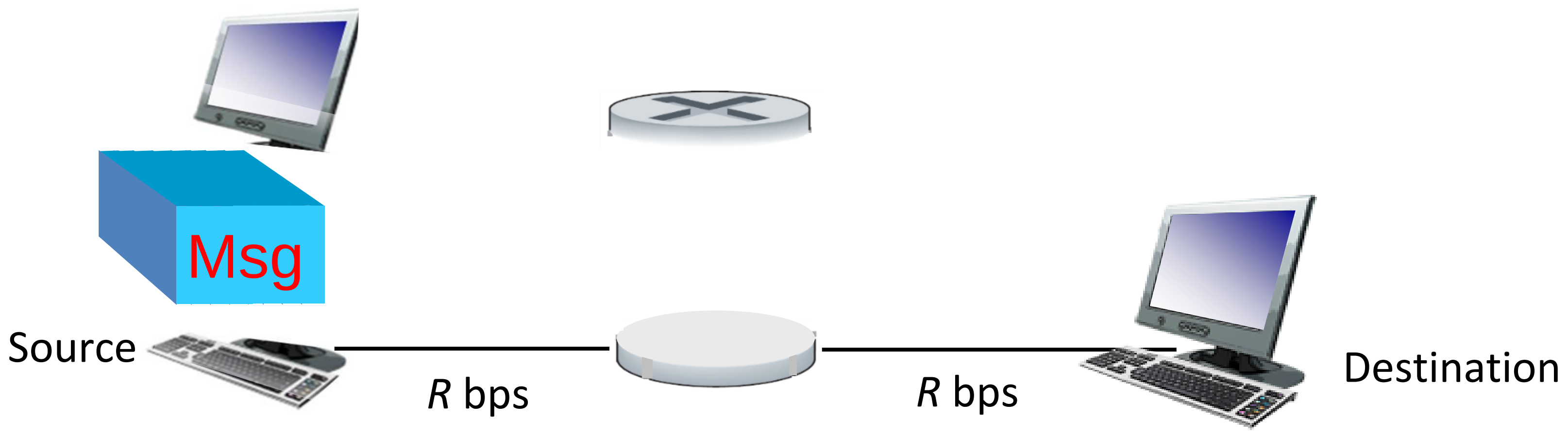


TDM



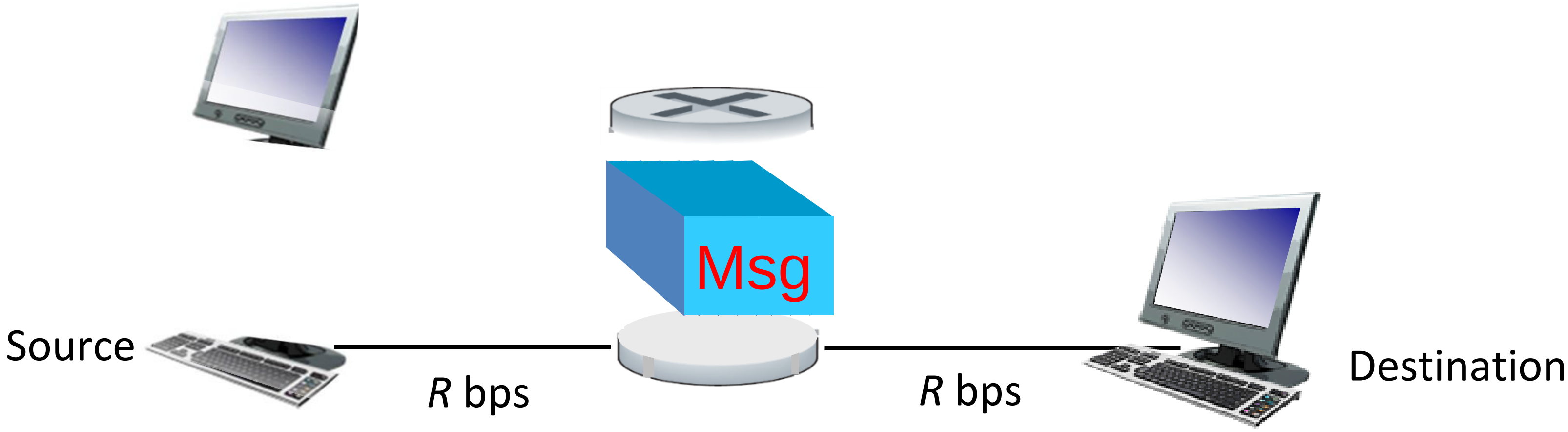
Message Switching

- Message(报文)：Overall information from source or application.
  - e.g.： a file



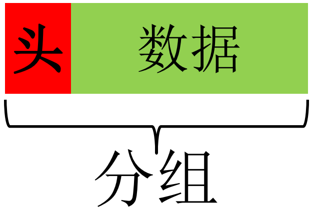
Message Switching

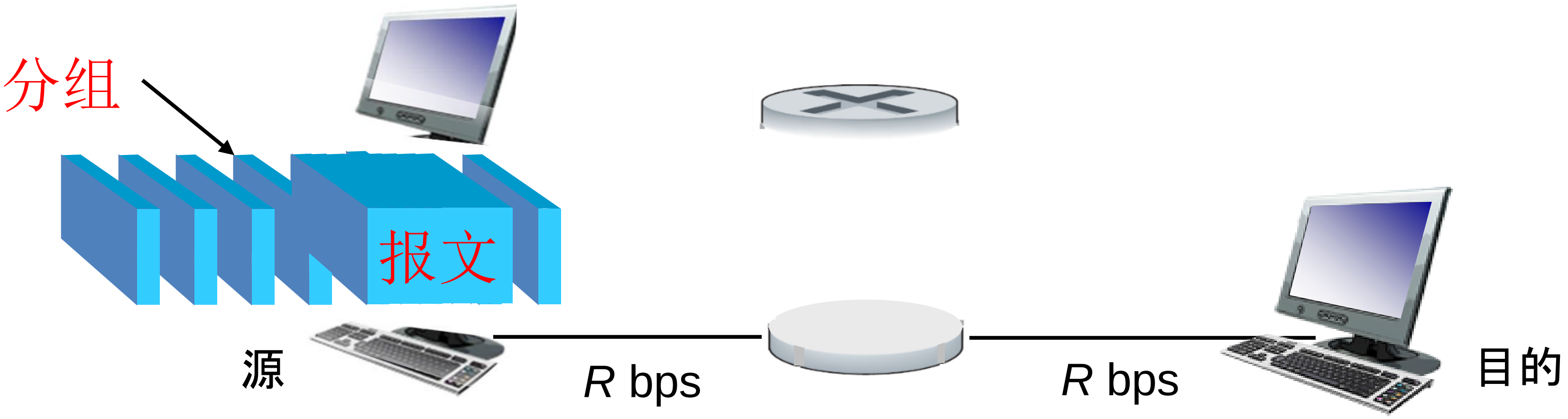
- Message(报文)： Overall information from source or application.
  - e.g.： a file



分组交换（package switching）

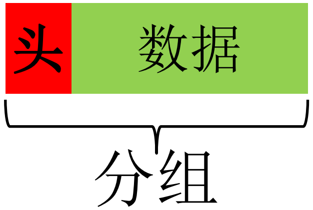
- 分组：报文分拆出来的一系列相对较小的数据包
- 分组交换需要报文的拆分与重组
- 产生额外开销

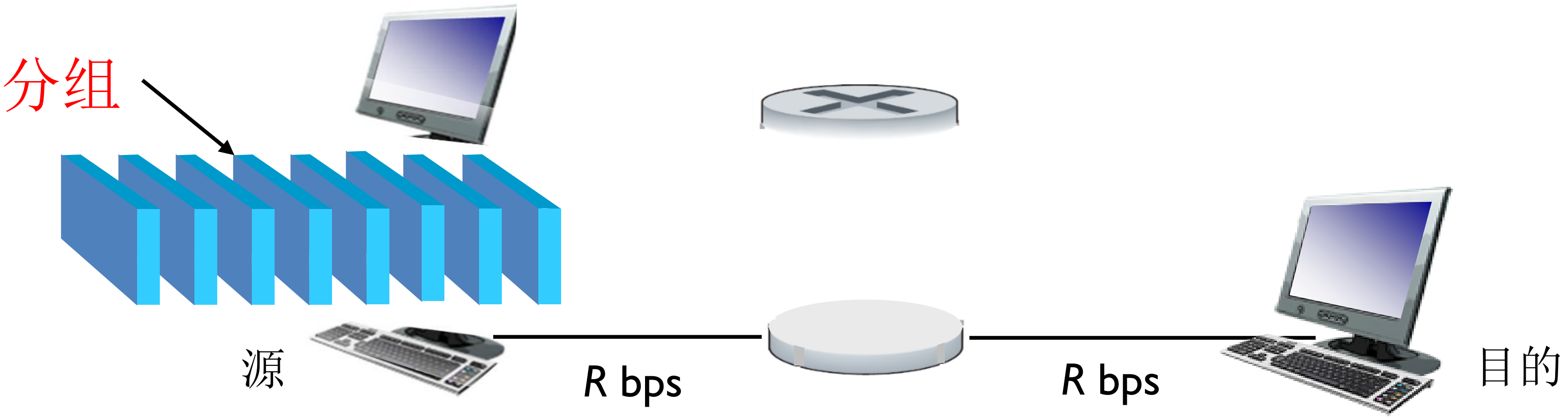




分组交换（package switching）

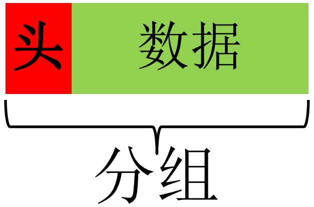
- 分组：报文分拆出来的一系列相对较小的数据包
- 分组交换需要报文的拆分与重组
- 产生额外开销

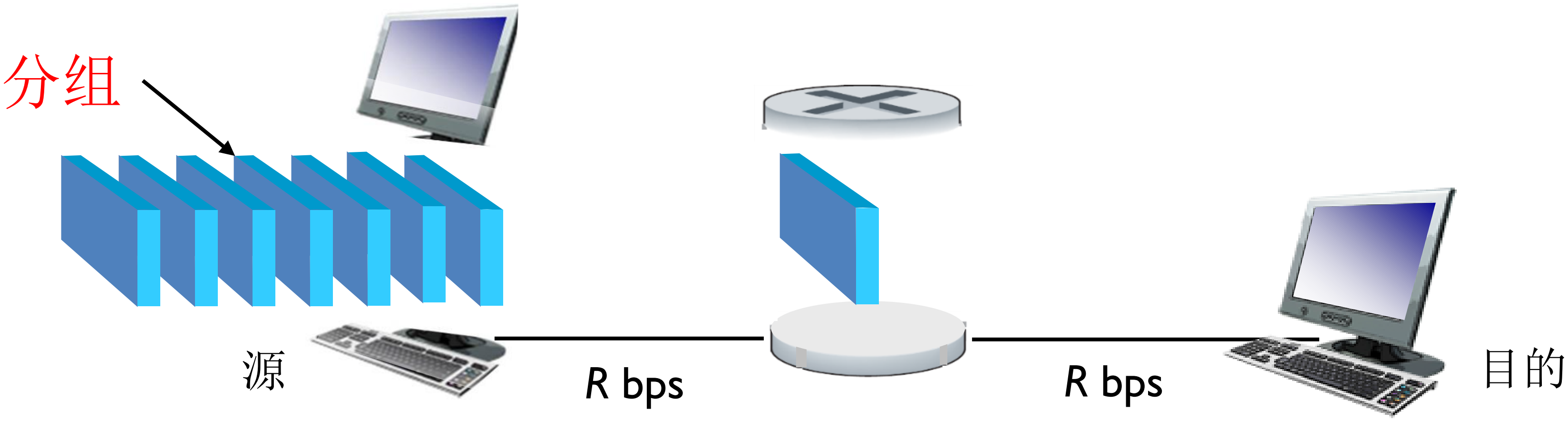




分组交换（package switching）

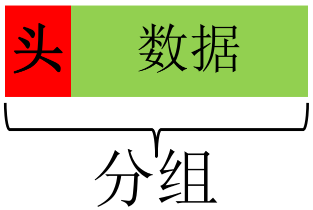
- 分组：报文分拆出来的一系列相对较小的数据包
- 分组交换需要报文的拆分与重组
- 产生额外开销

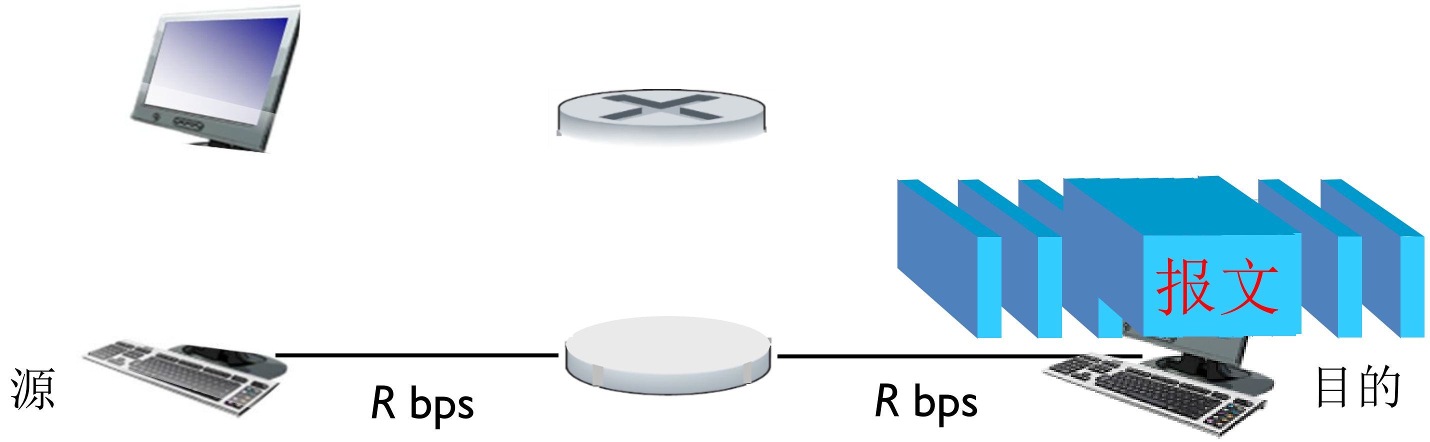




分组交换（package switching）

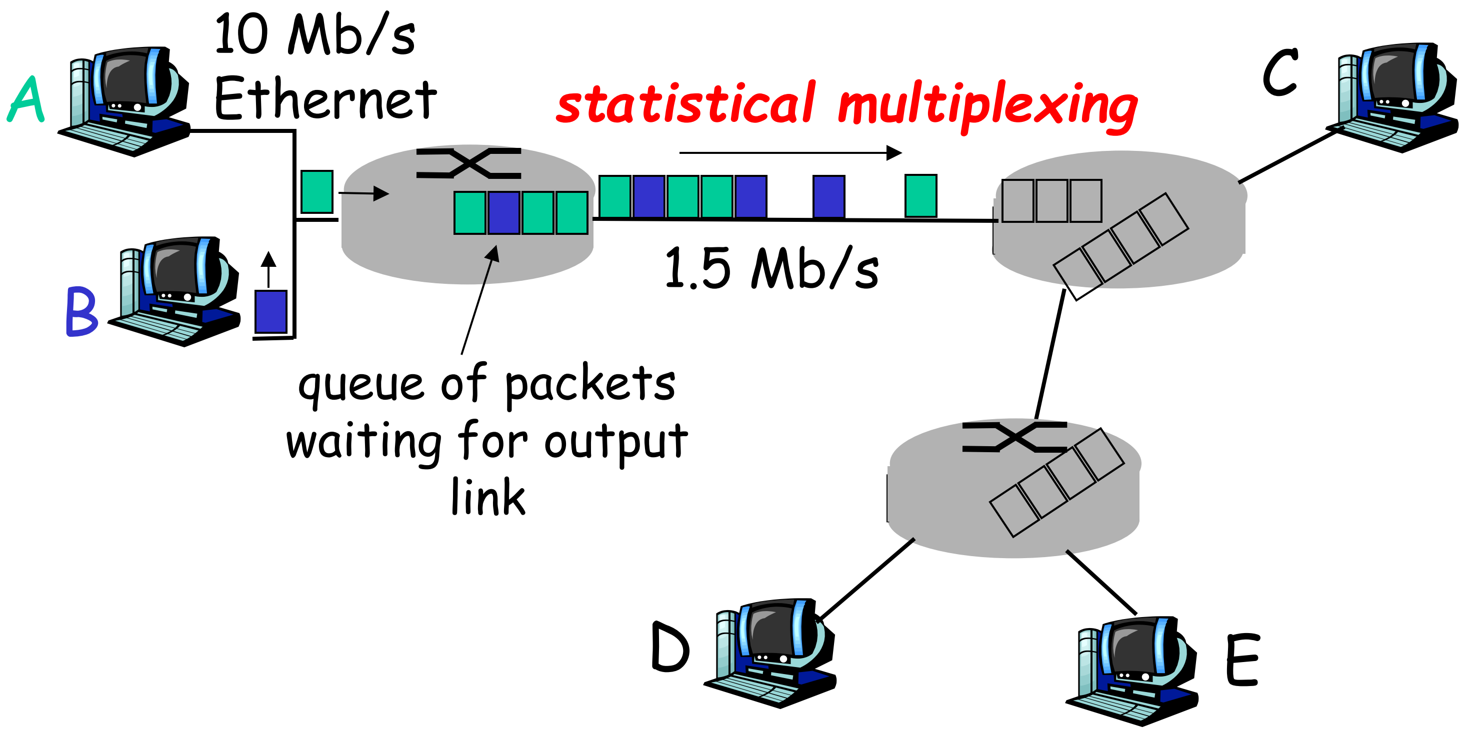
- 分组：报文分拆出来的一系列相对较小的数据包
- 分组交换需要报文的拆分与重组
- 产生额外开销






开点工作室 / 编著
横扫
Offer
程序员招聘真题详解700题
Find Job
清华大学出版社

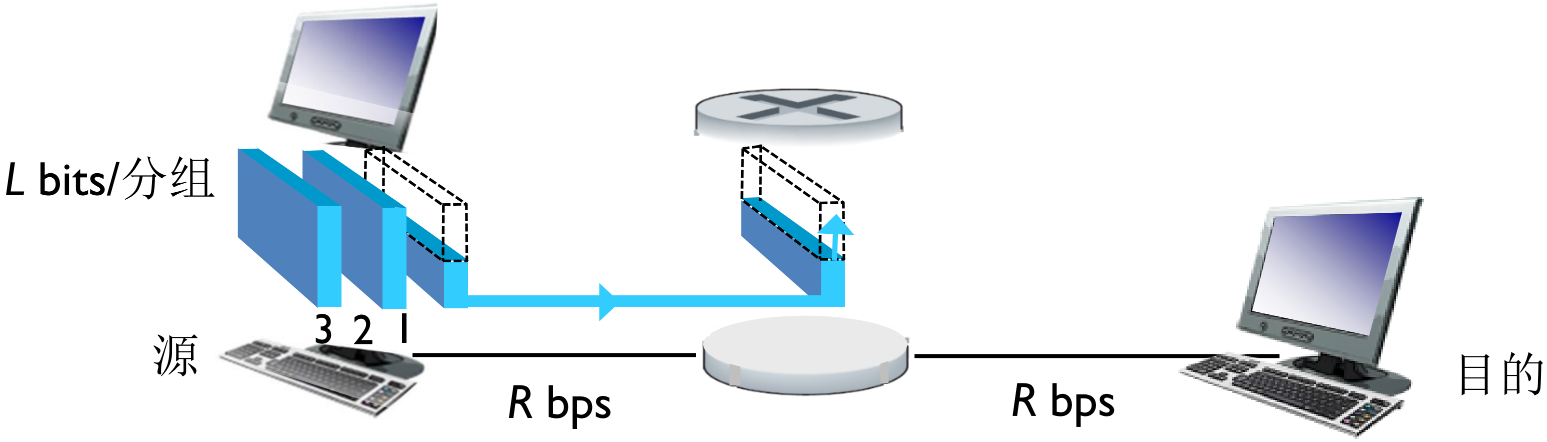
Packet Switching: Statistical Multiplexing



Sequence of A & B packets does not have fixed pattern, shared on demand ➔ ***statistical multiplexing***.

TDM: each host gets same slot in revolving TDM frame.

存储-转发（store-and-forward）



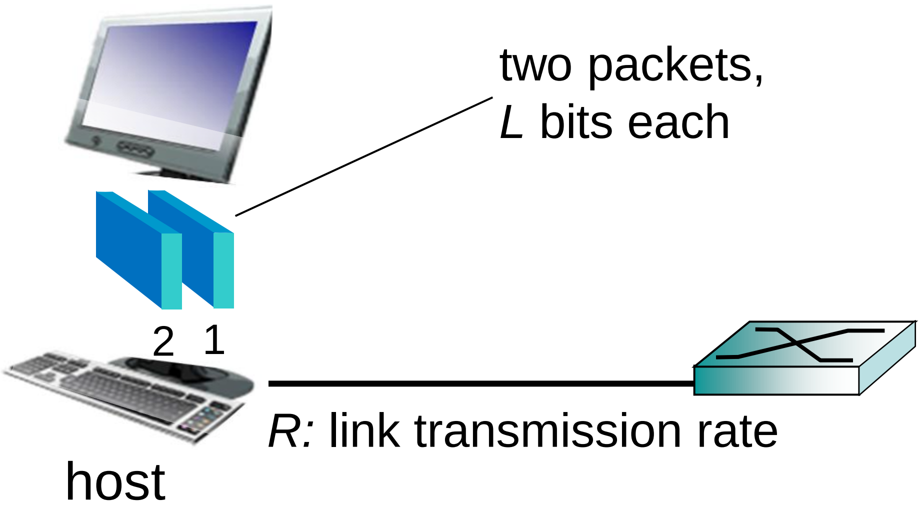
- 报文交换与分组交换均采用存储-转发交换方式
- 区别：
  - 报文交换以完整报文进行“存储-转发”
  - 分组交换以较小的分组进行“存储-转发”

- 哪种交换方式更好呢？

Host: sends *packets* of data

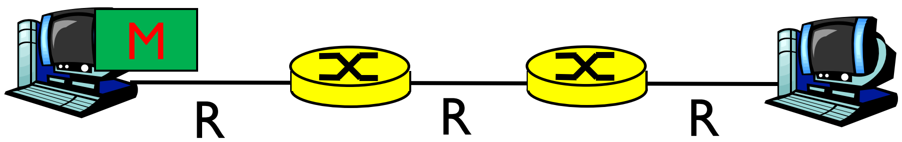
host sending function:

- takes application message
- breaks into smaller chunks, known as *packets*, of length *L* bits
- transmits packet into access network at *transmission rate R*
  - link transmission rate, aka link *capacity, aka link bandwidth*



packet transmission delay = time needed to transmit *L*-bit packet into link = $\frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$

报文交换 vs 分组交换?

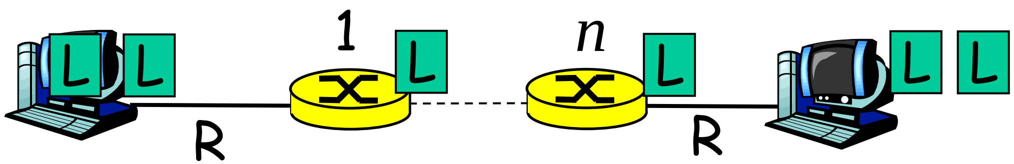


- 报文交换:
  - 报文长度为M bits
  - 链路带宽为R bps
  - 每次传输报文需要M/R秒
- 分组交换:
  - 报文被拆分为多个分组
  - 分组长度为L bits
  - 每个分组传输时延为L/R秒

例:

- M=7.5 Mbits，L=1500bits
  - M=5000L
- R = 1.5 Mbps
- 报文交换:
  - 报文交付时间=? sec
- 分组交换:
  - 报文交付时间=? sec

Packet-switching: delay

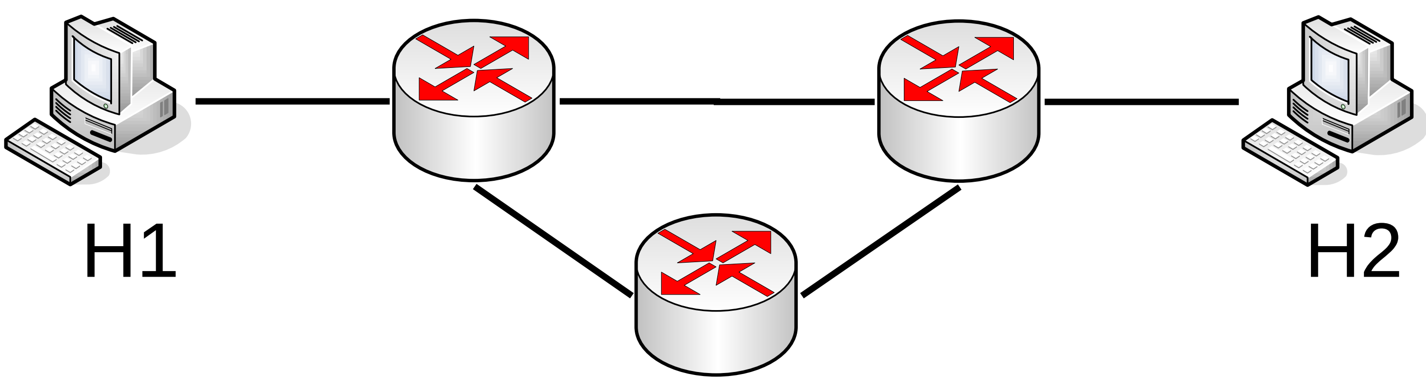


- Message: M bits
- Link bandwidth: R bps
- Packet size: L bits
- Hops #: *h*
- Routers #： *n*

$$T=M/R+(h-1)L/R$$
$$=M/R+nL/R$$

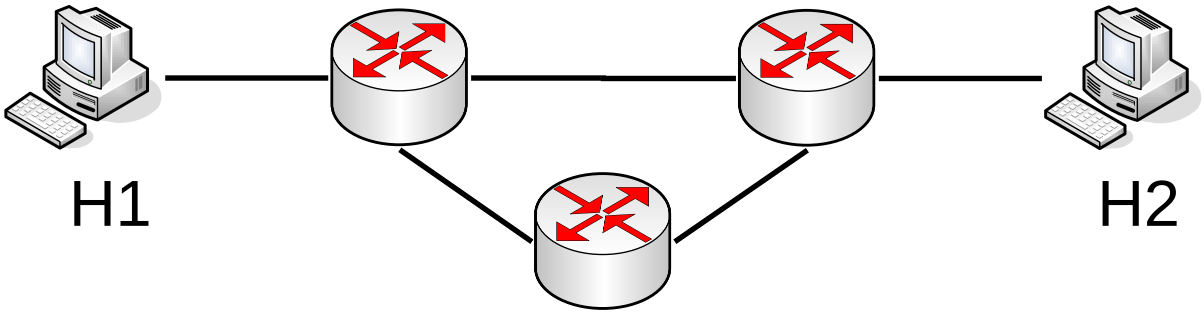
例题1.1

- 在下图所示的采用“存储-转发”方式的分组交换网络中，所有链路的数据传输速率为100 Mbps，分组大小为1 000 B，其中分组头大小为20 B。若主机H1向主机H2发送一个大小为980 000 B的文件，则在不考虑分组拆装时间和传播延迟的情况下，从H1发送开始到H2接收完为止，需要的时间至少是？



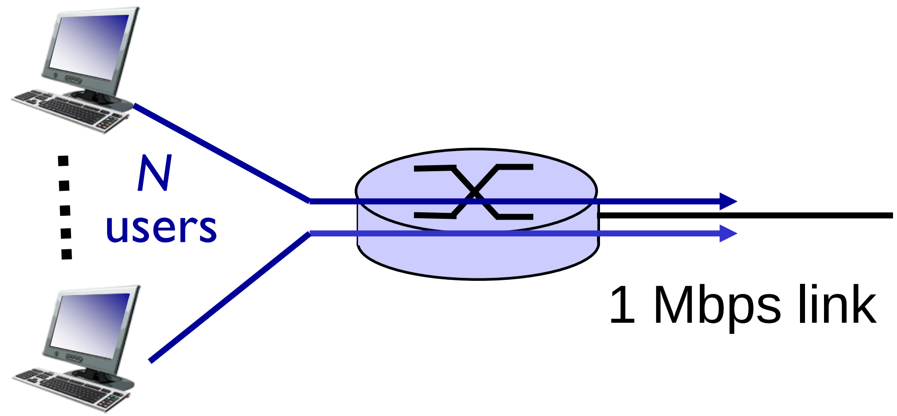
例题1.1

- 【解】980 000 B大小的文件需要分1000个分组，每个分组1 000 B。H1发送整个文件需要的传输延迟为(980 000+20*1000)*8/100 000 000=80ms；根据路由选择基本原理，所有数据分组应该经过两个路由器的转发，所以再加上最后一个分组的两次转发的传输延迟，即2*1000*8/100 000 000=0.16ms。所以，H2收完整个文件至少需要80+0.16=80.16ms。



Packet switching versus circuit switching

- example:
- 1 Mb/s link
- each user:
  - ✓ 100 kb/s when "active"
  - ✓ active 10% of time



- *circuit-switching:*
  - ✓ 10 users
- *packet switching:*
  - ✓ with 35 users, probability > 10 active at same time is less than .0004

Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

packet switching allows more users to use network!

Packet switching versus circuit switching

Is packet switching a "slam dunk winner?"

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- *Q:* How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem (chapter 7)

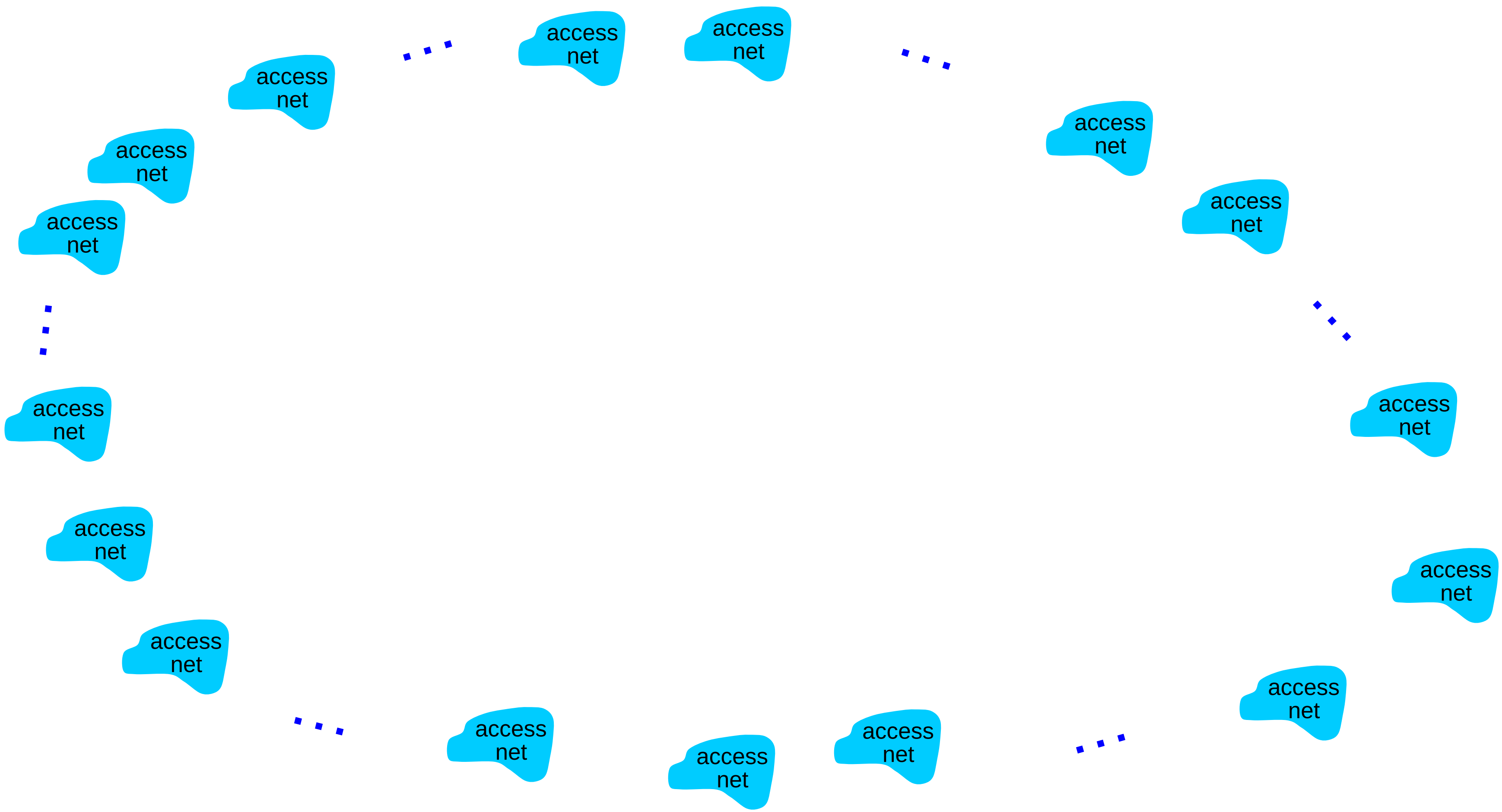
Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Internet structure: network of networks

- End systems connect to Internet via access ISPs (Internet Service Providers)
  - Residential, company and university ISPs
- Access ISPs in turn must be interconnected.
  - So that any two hosts can send packets to each other
- Resulting network of networks is very complex
  - Evolution was driven by economics and national policies
- Let's take a stepwise approach to describe current Internet structure

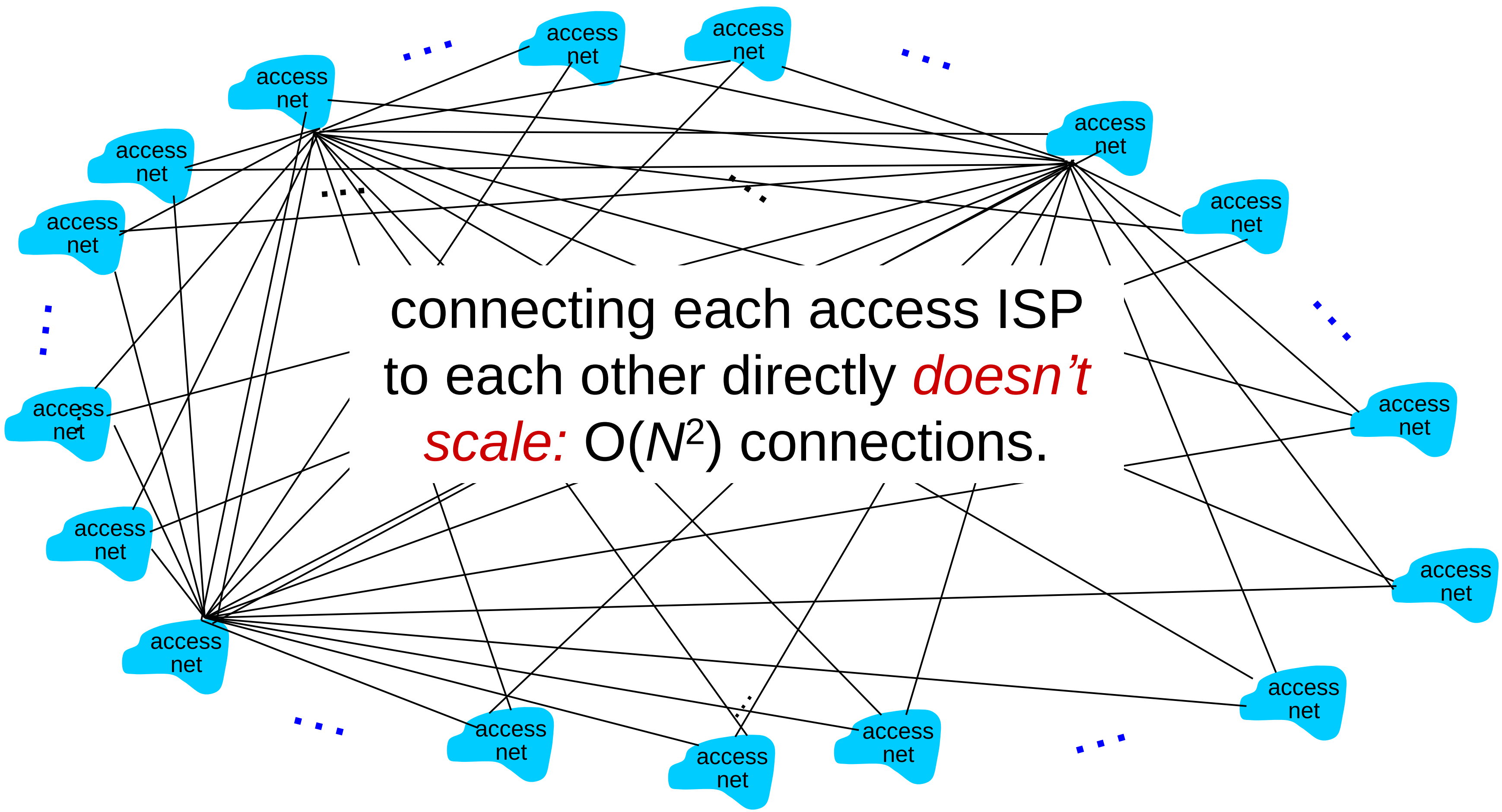
Internet structure: network of networks

Question: given *millions* of access ISPs, how to connect them together?



Internet structure: network of networks

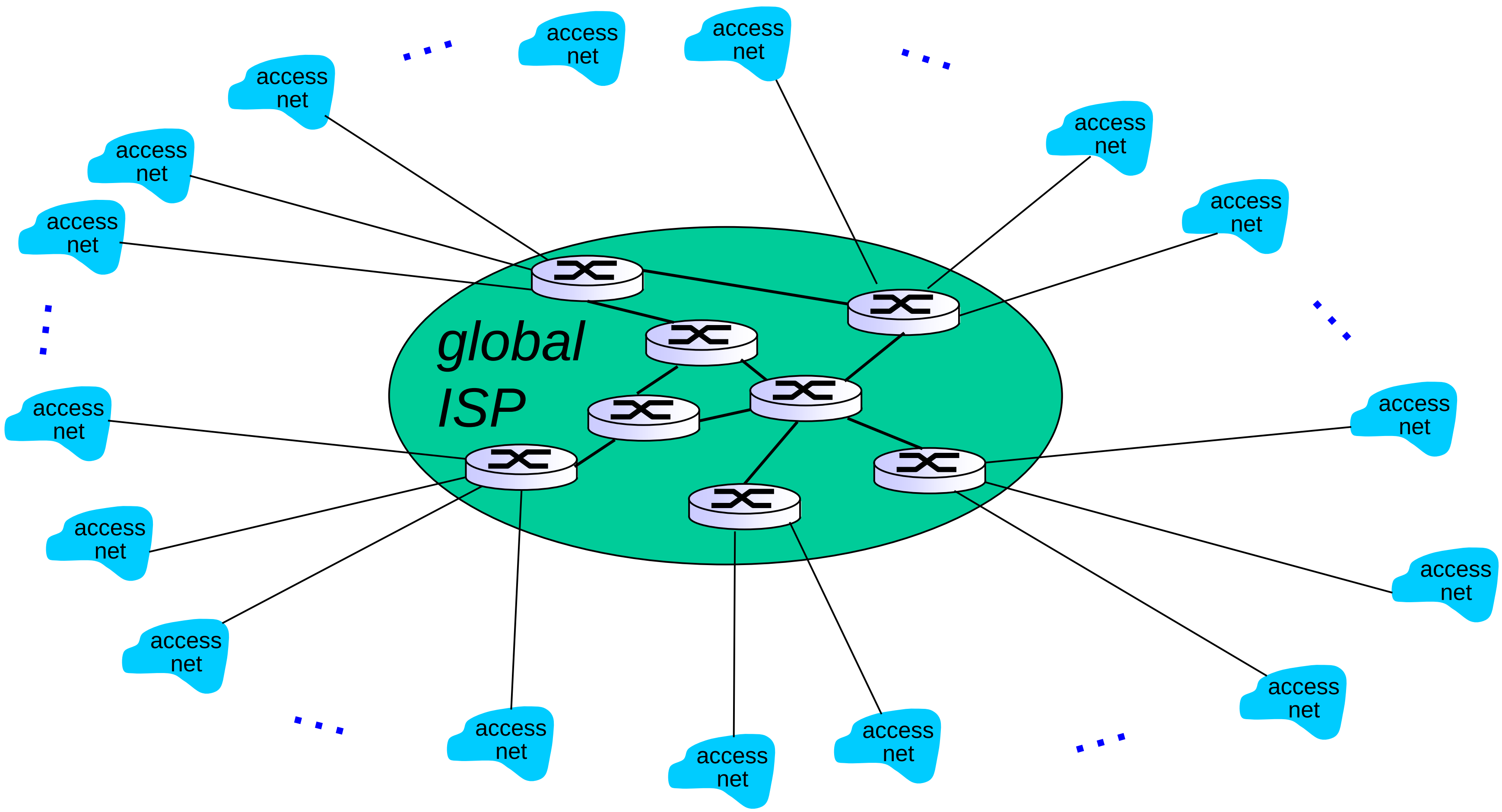
Option: connect each access ISP to every other access ISP?



connecting each access ISP to each other directly *doesn't scale:* $O(N^2)$ connections.

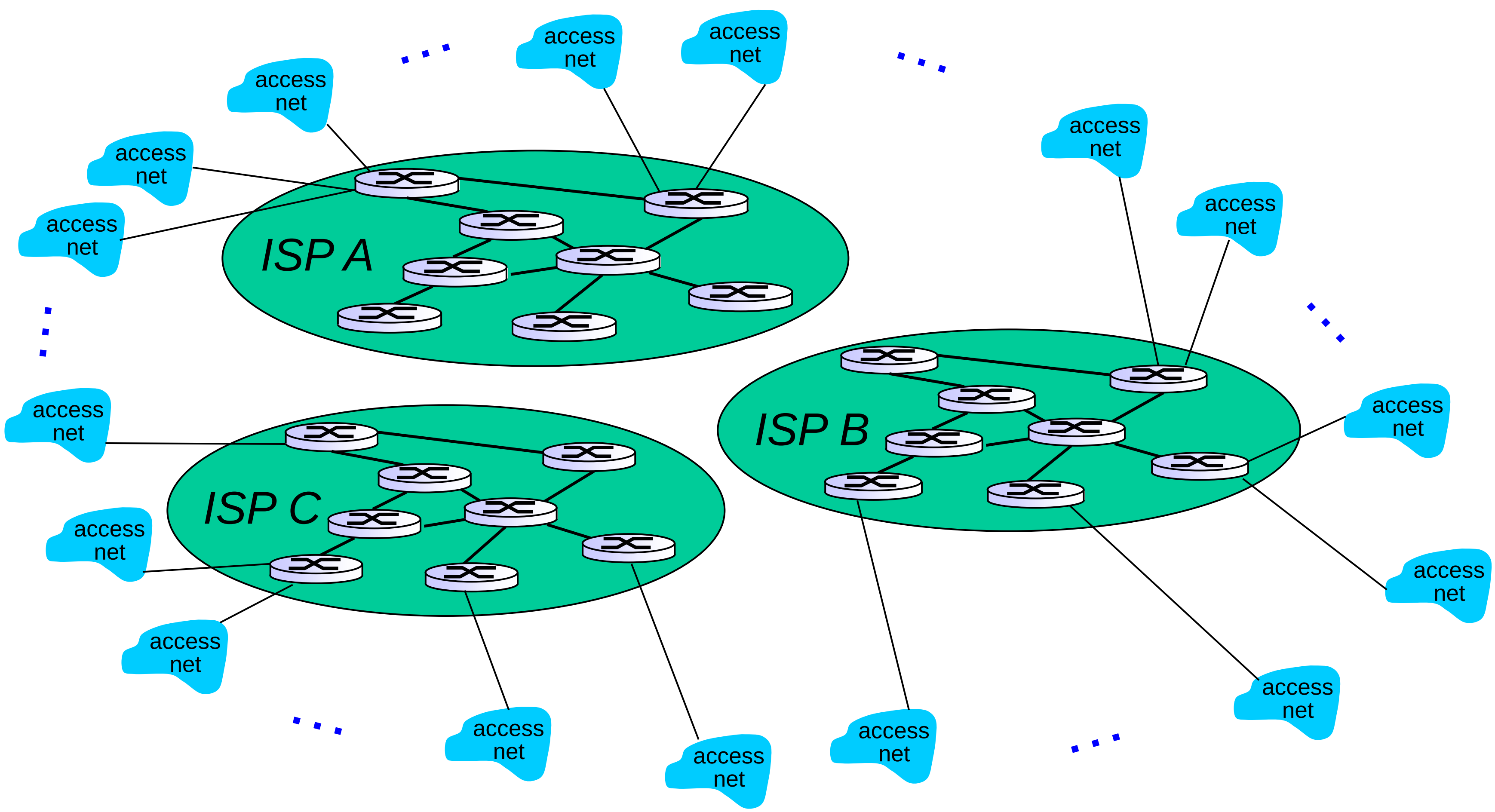
Internet structure: network of networks

Option: *connect each access ISP to a global transit ISP?* *Customer* *and* *provider* *ISPs have economic agreement.*



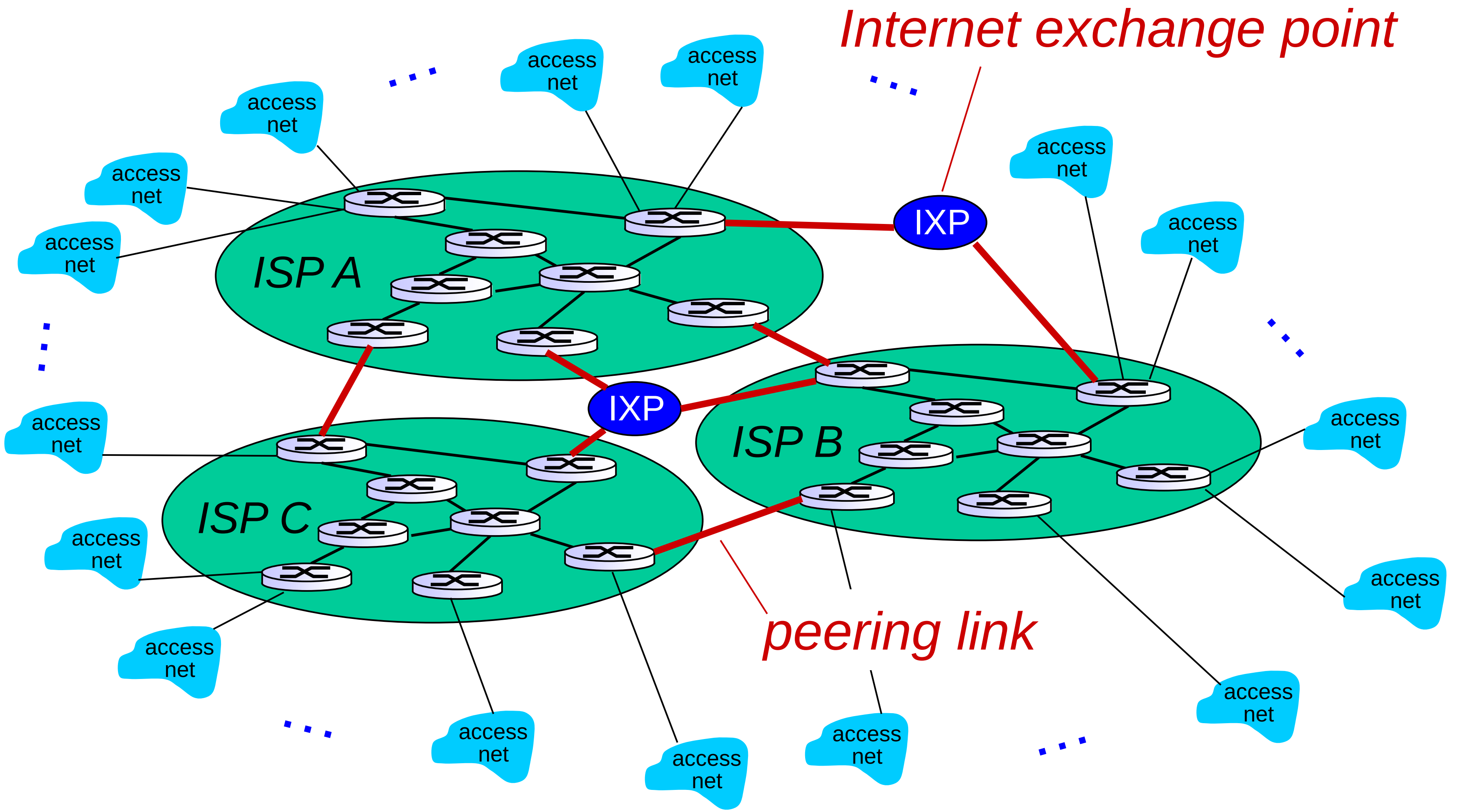
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors ….



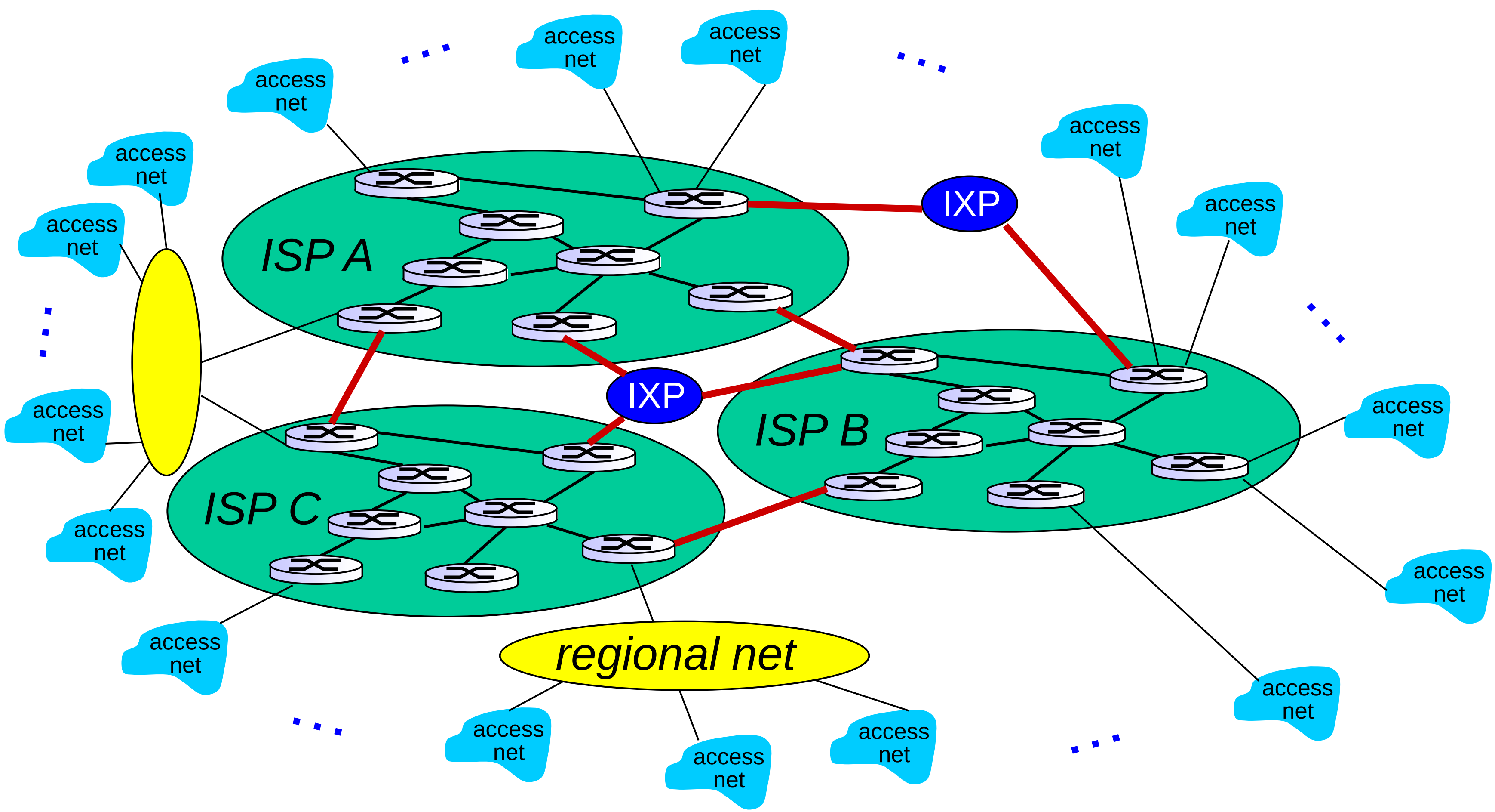
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors …. which must be interconnected



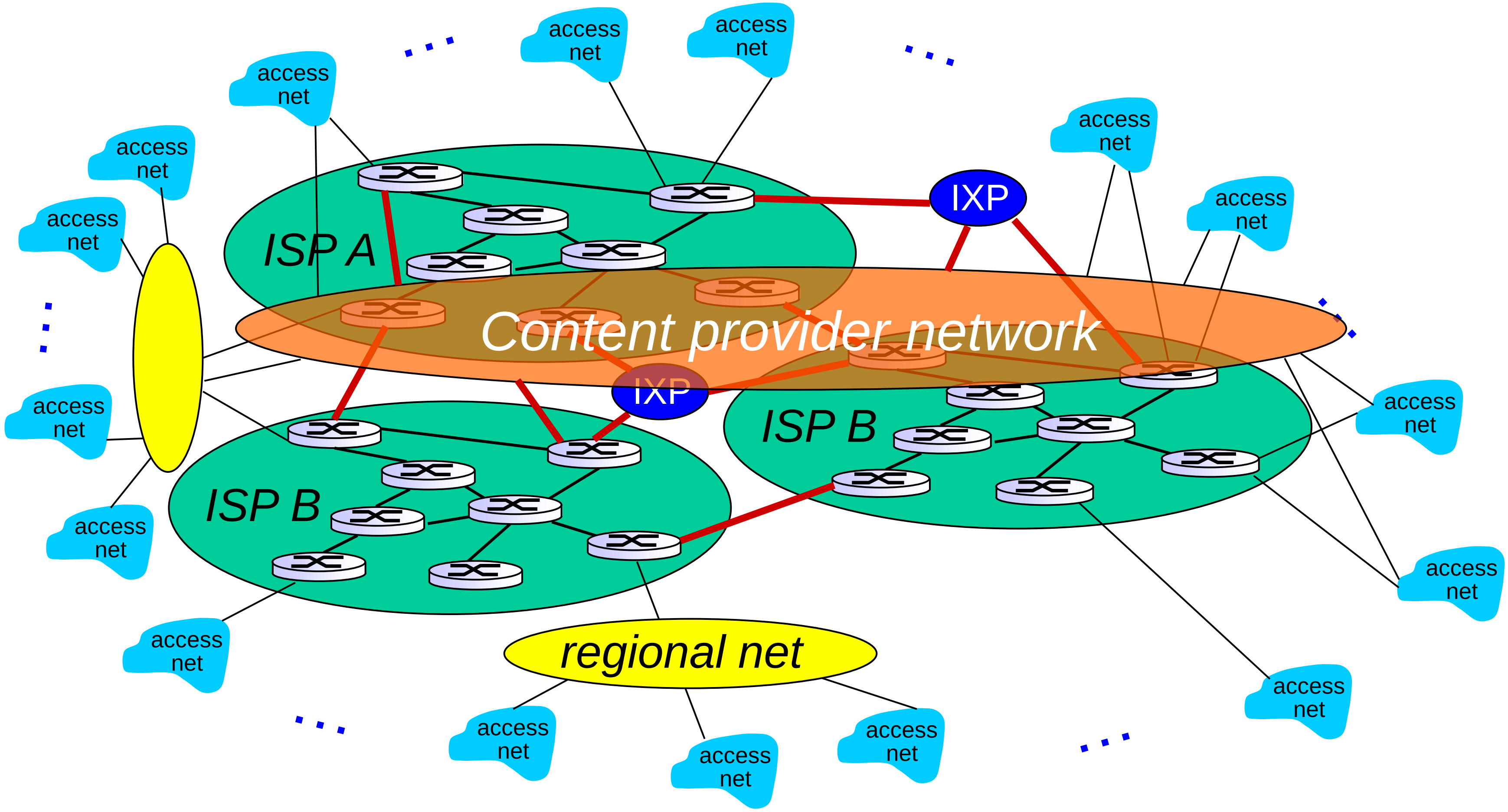
Internet structure: network of networks

… and regional networks may arise to connect access nets to ISPS

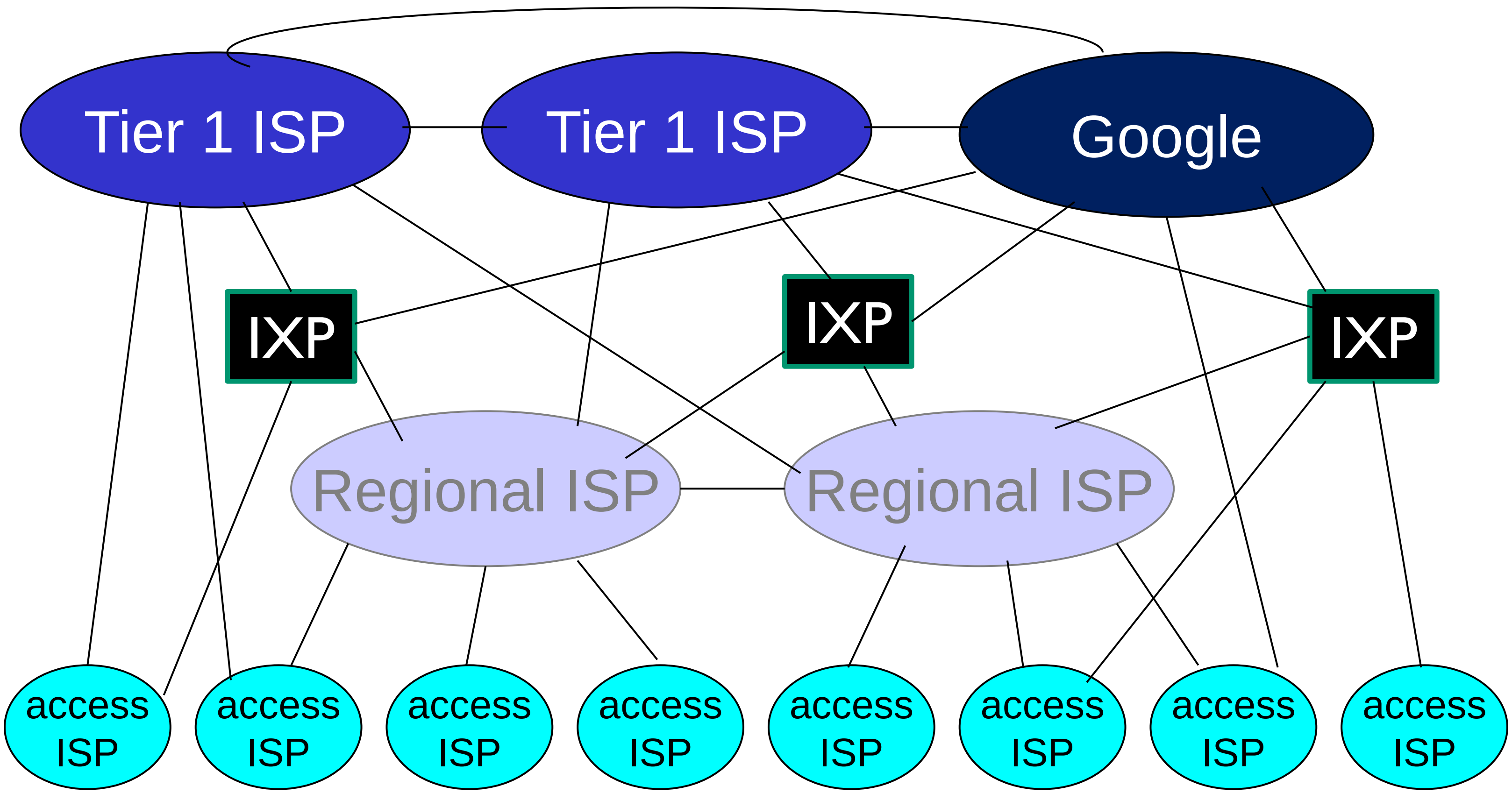


Internet structure: network of networks

… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users

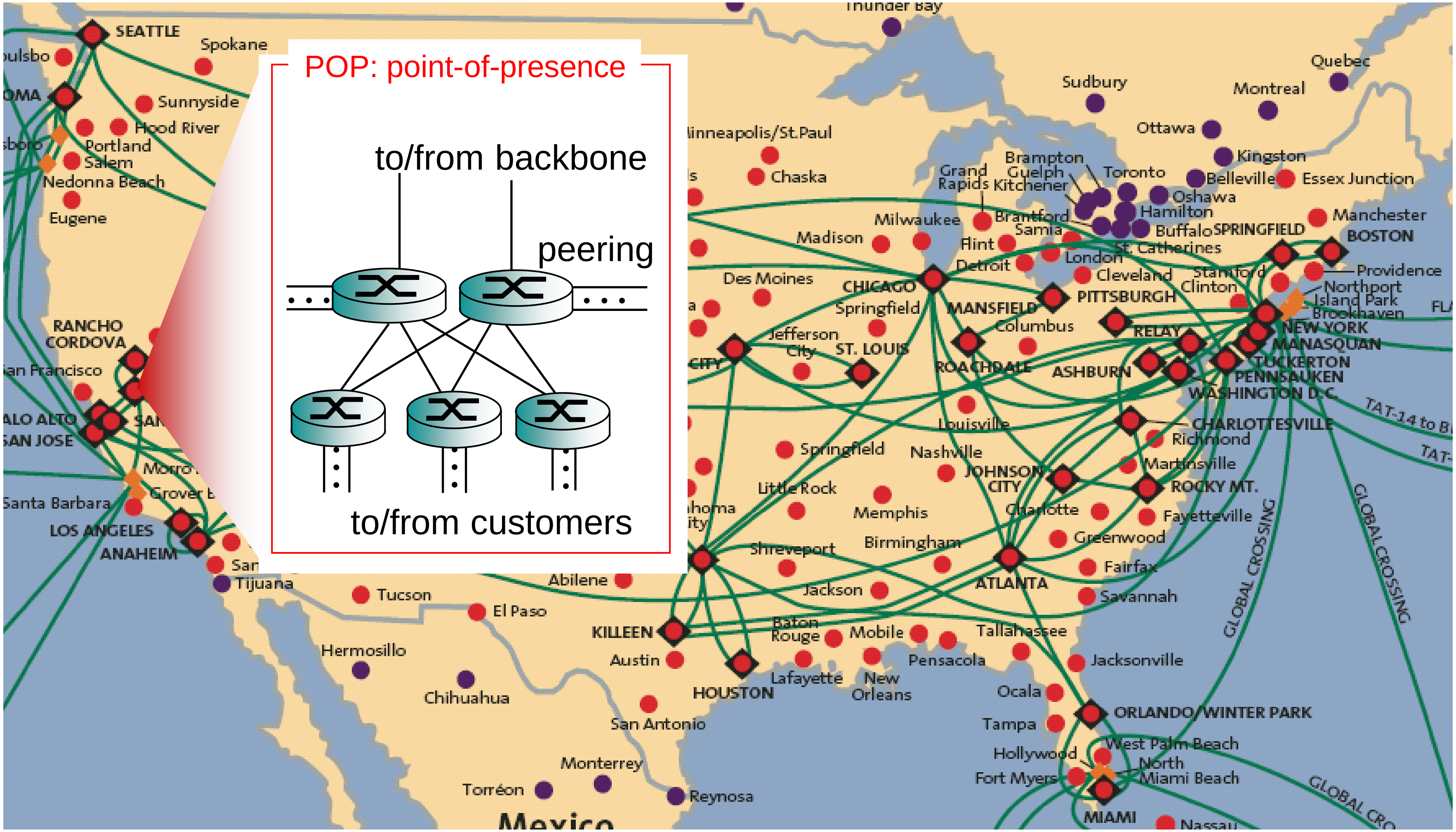


Internet structure: network of networks



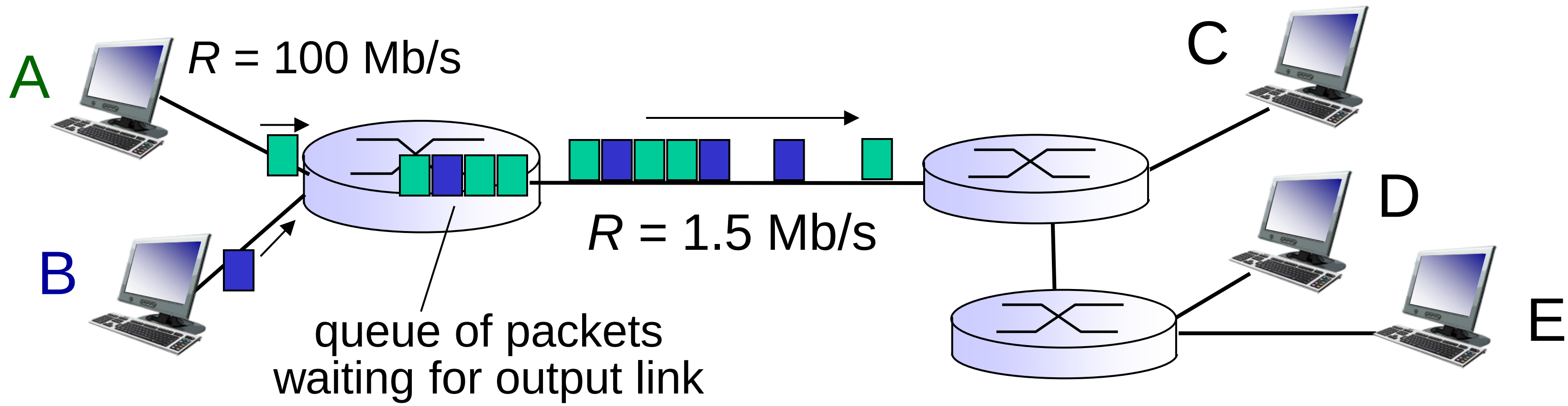
- at center: small # of well-connected large networks
  - "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - content provider network (e.g, Google): private network that connects it's data centers to Internet, often bypassing tier-1, regional ISPs

Tier-1 ISP: e.g., Sprint



Chapter 1: roadmap

Packet Switching: queueing delay, loss



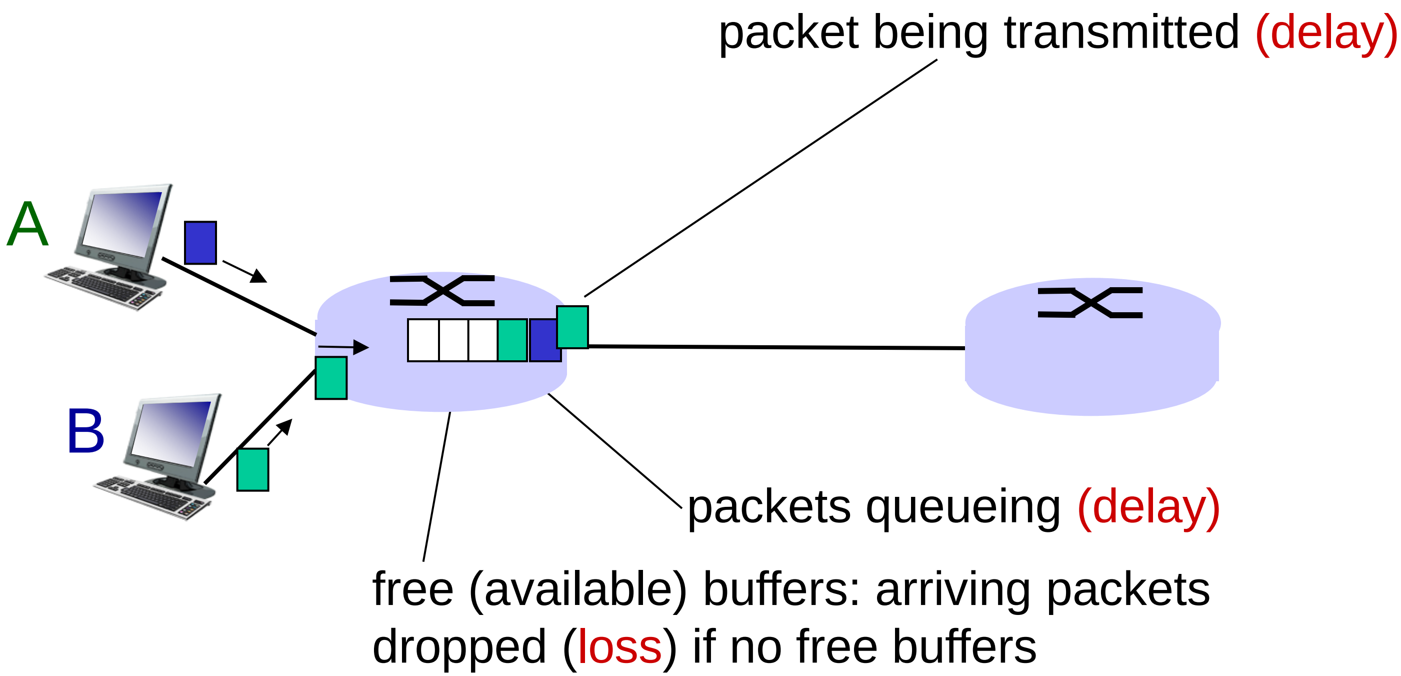
queuing and loss:

- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

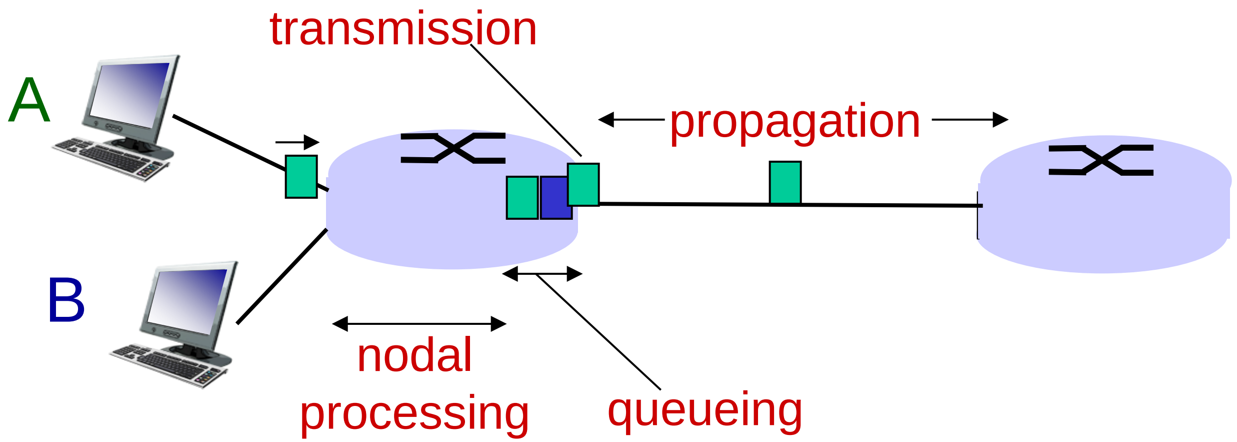
How do loss and delay occur?

packets *queue* in router buffers

- packet arrival rate to link (temporarily) exceeds output link capacity
- packets queue, wait for turn



Four sources of packet delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

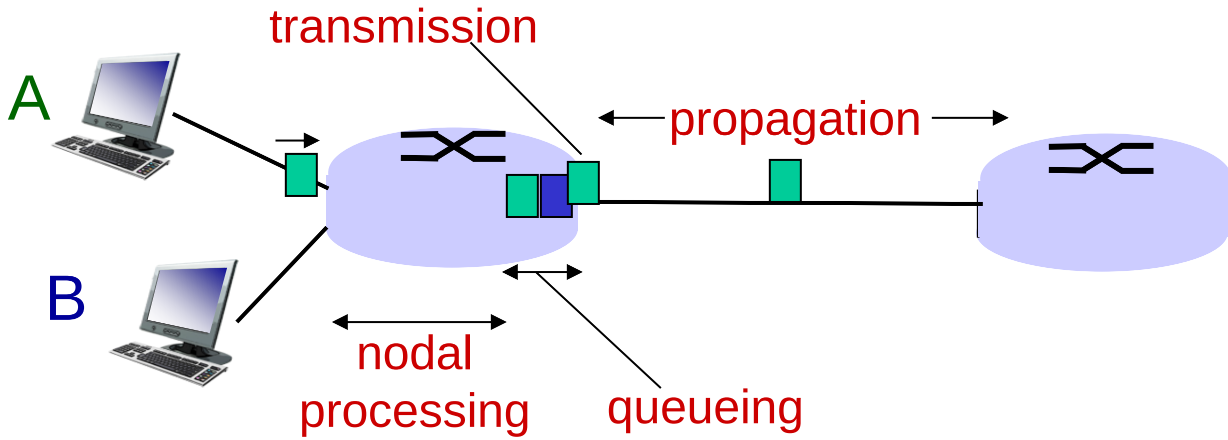
d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Four sources of packet delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:

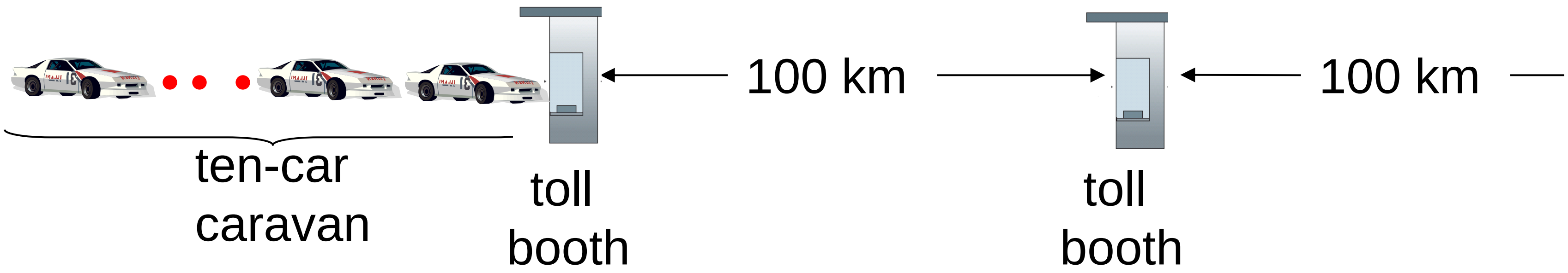
- *L*: packet length (bits)
- *R*: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay:

- *d*: length of physical link
- *s*: propagation speed in medium (~2x10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

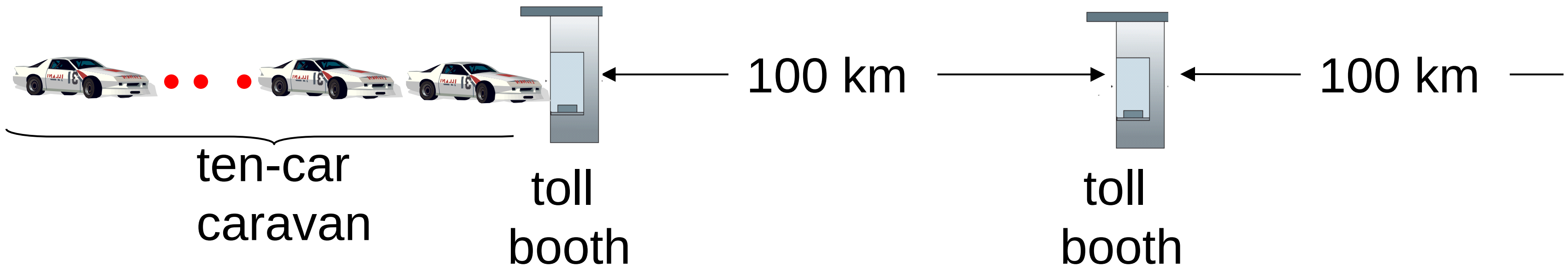
Caravan analogy



- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car~bit; caravan ~ packet
- Q: How long until caravan is lined up before 2nd toll booth?

- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- *A:* 62 minutes

Caravan analogy (more)



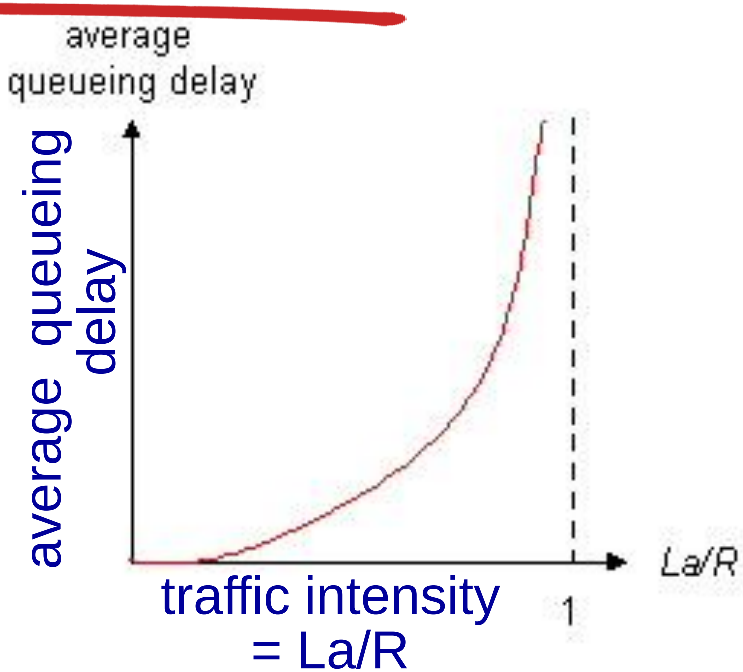
- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?

  - *A: Yes!* after 7 min, 1st car arrives at second booth; three cars still at 1st booth.

Queueing delay (revisited)

- *R*: link bandwidth (bps)
- *L*: packet length (bits)
- a: average packet arrival rate

traffic intensity = La/R



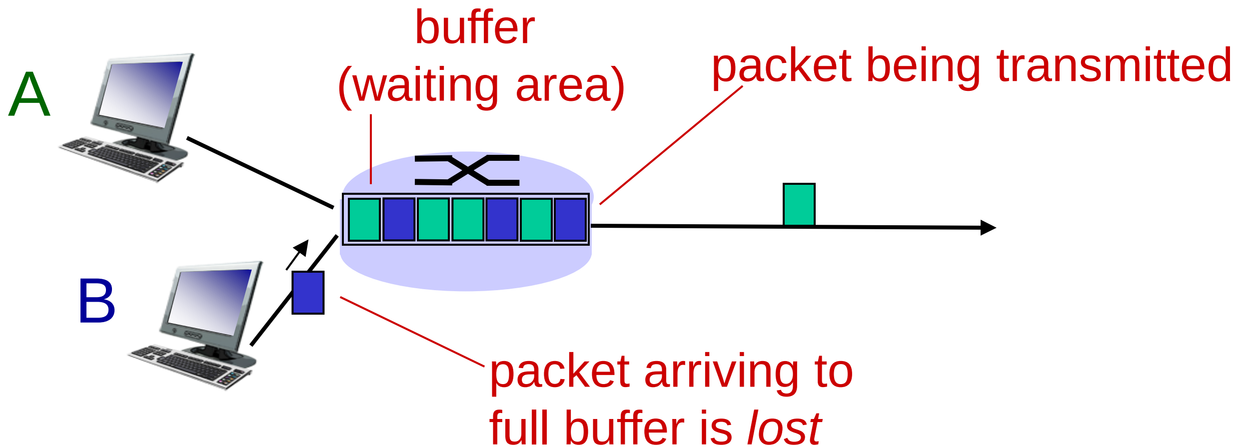
- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving than can be serviced, average delay infinite!

La/R ~ 0

La/R -> 1

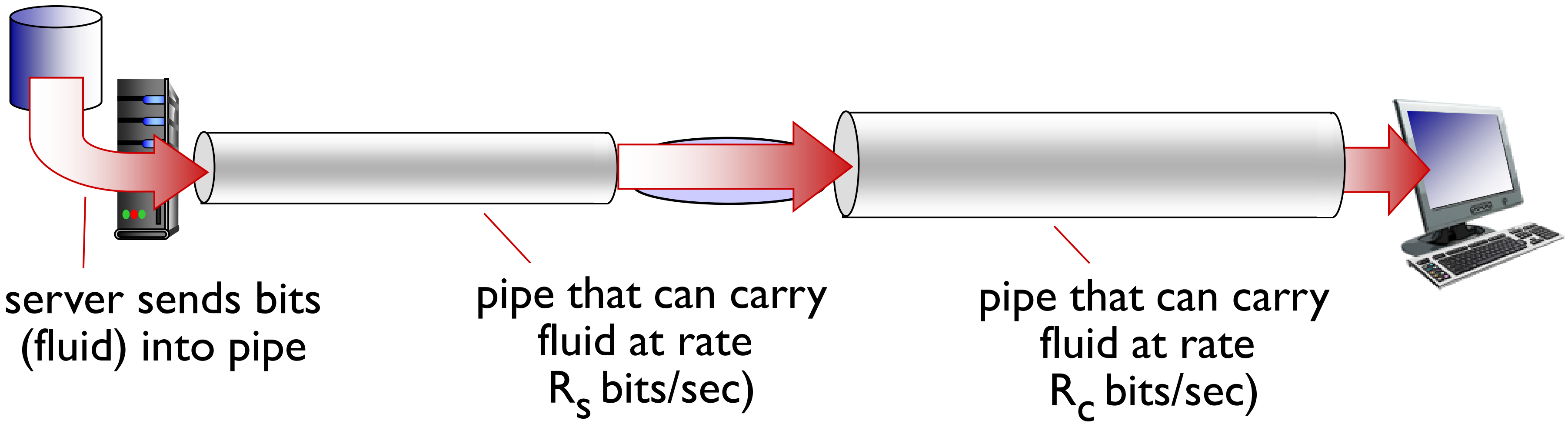
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



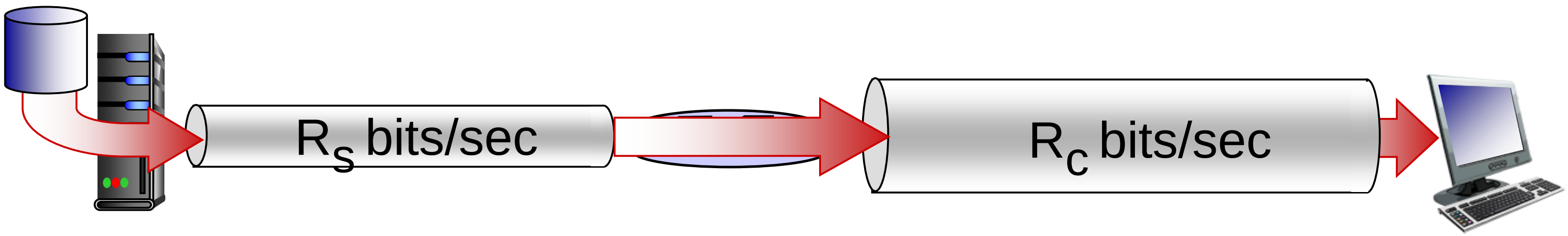
Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

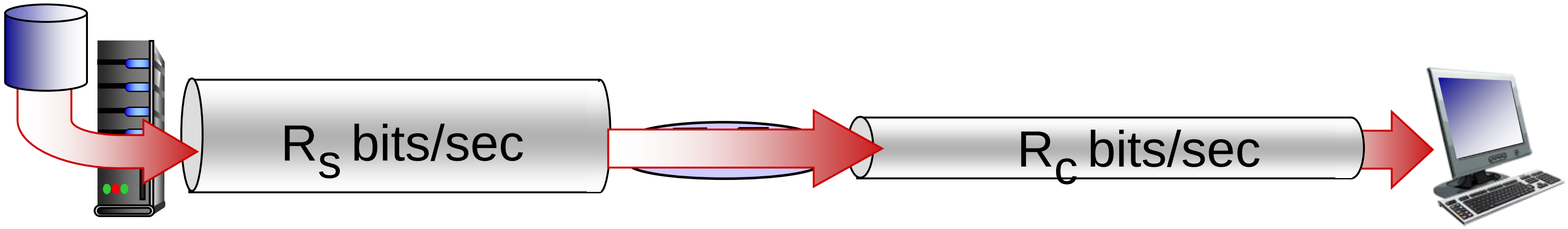


Throughput (more)

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?

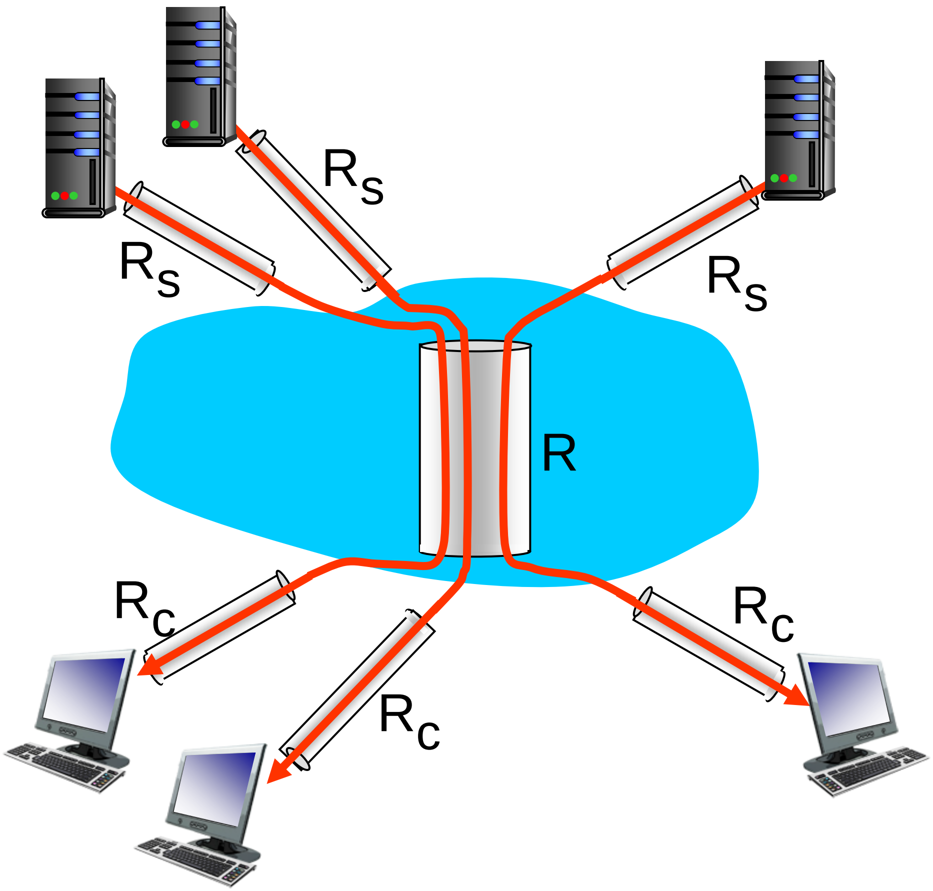


bottleneck link

link on end-end path that constrains end-end throughput

Throughput: Internet scenario

- per-connection end-end throughput: $\min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck



10 connections (fairly) share backbone bottleneck link R bits/sec

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 history

Protocol "layers"

Networks are complex,
with many "pieces":

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of *organizing* structure of network?

…. or at least our discussion of networks?

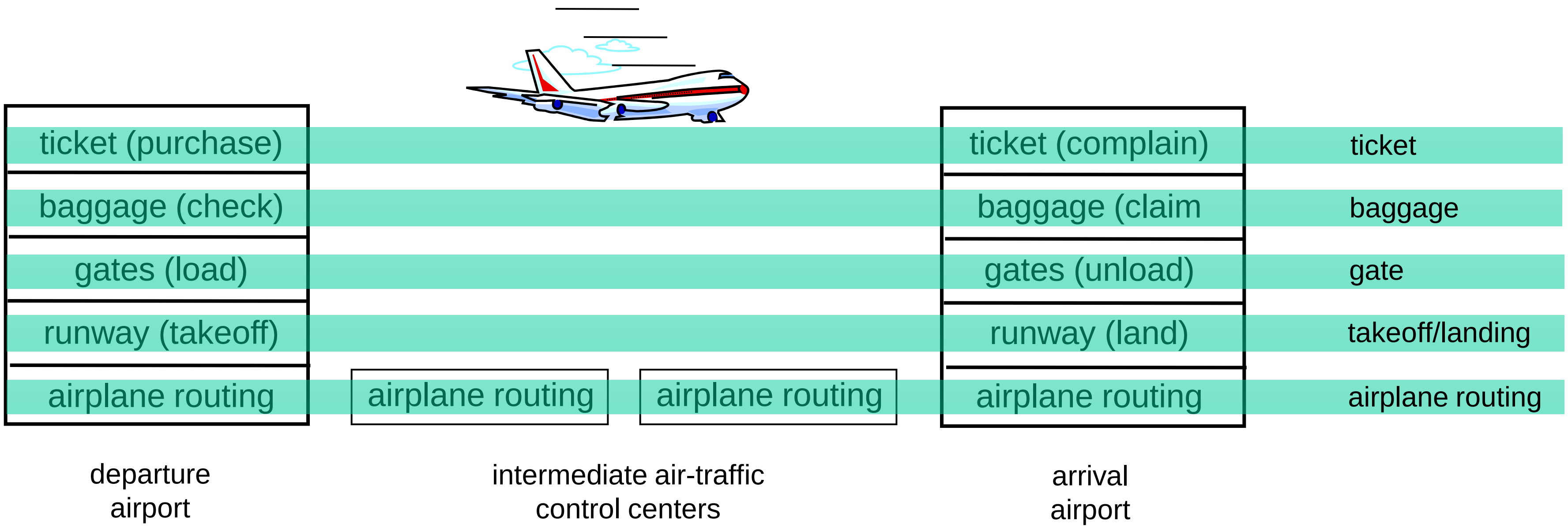
Organization of air travel

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing

- a series of steps

Layering of airline functionality



layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

Why layering?

dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

OSI参考模型

- 开放系统互连 (OSI)参考模型是由国际标准化组织 (ISO) 于1984年提出的分层网络体系结构模型
- 目的是支持异构网络系统的互联互通
- 异构系统互连的国际标准
- 理解网络通信过程的最佳学习工具（理论模型）
  - 理论成功，市场失败
- 7层（功能）,每层完成特定的网络功能

7: 应用层（Application）

6: 表示层（Presentation）

5: 会话层（Session）

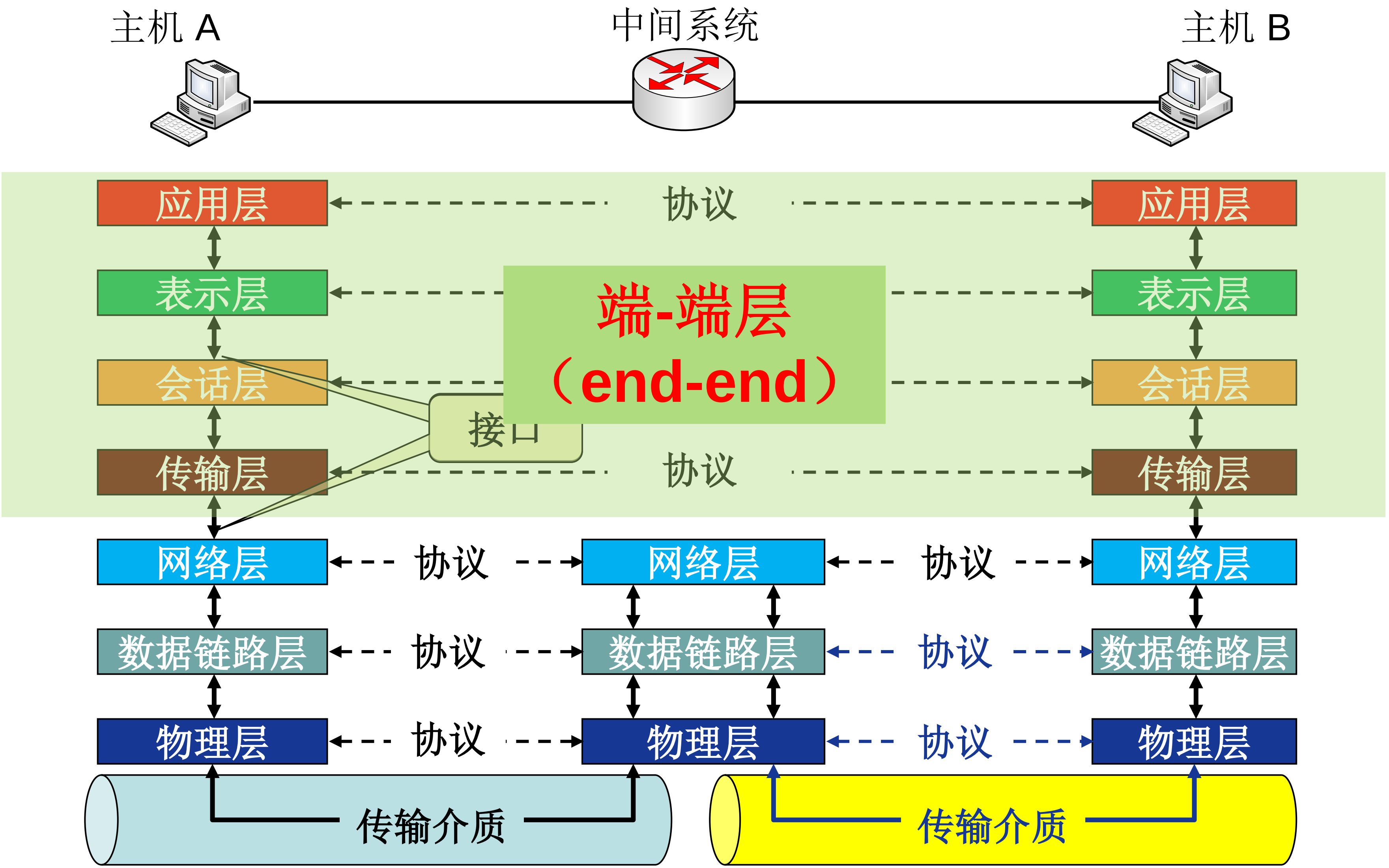
4: 传输层（Transport）

3: 网络层（Network）

2: 数据链路层（Data link）

1: 物理层（Physical）

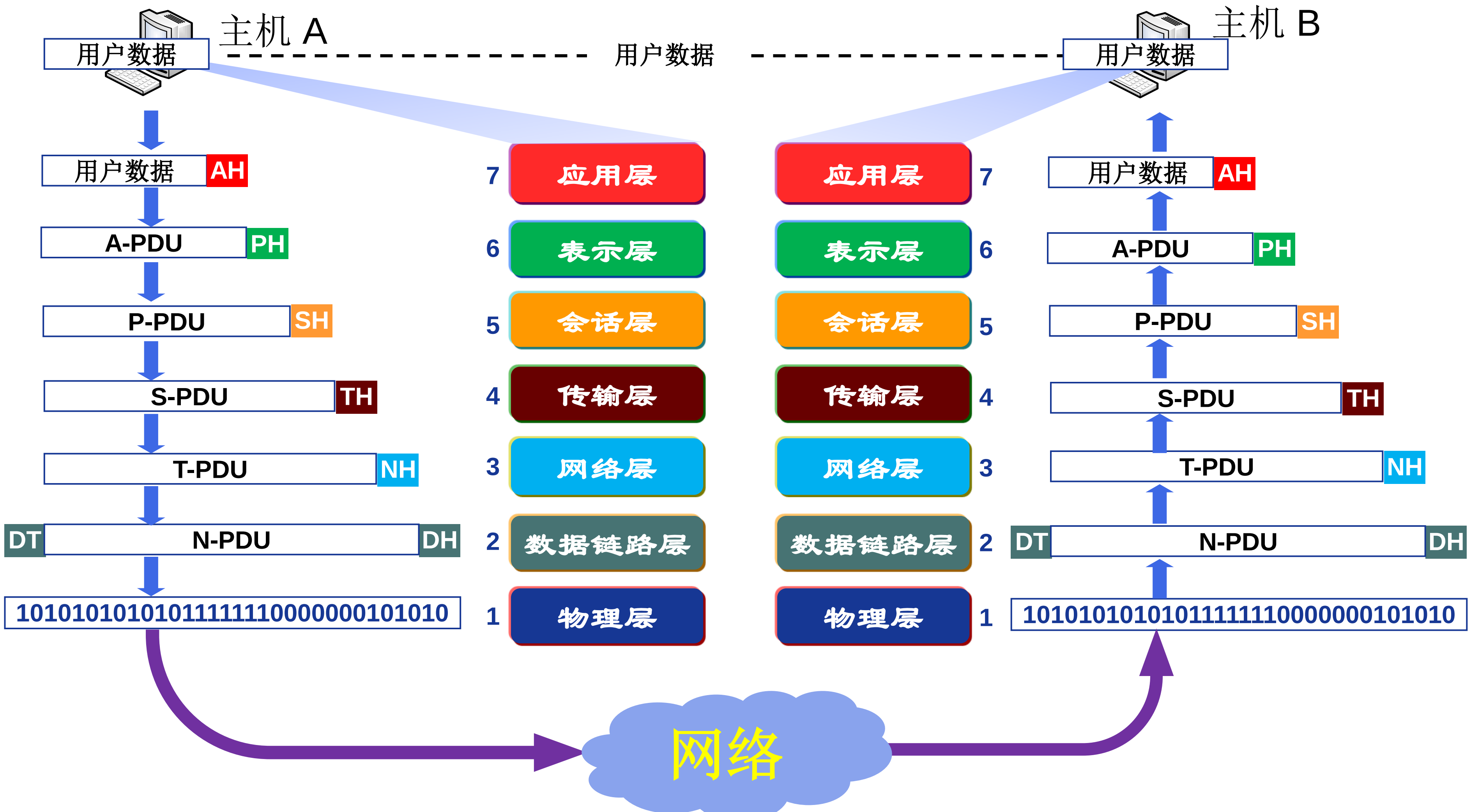
OSI参考模型解释的通信过程



层次网络体系结构

- 层层相叠，各司其职，依下奉上，彼此透明；自上而下，逐层封装，自下而上，逐层解封；同层同议，对等协商，实际传输，下层帮忙；底层相邻，高层相望，最终通信，物理担当。

OSI参考模型数据封装与通信过程



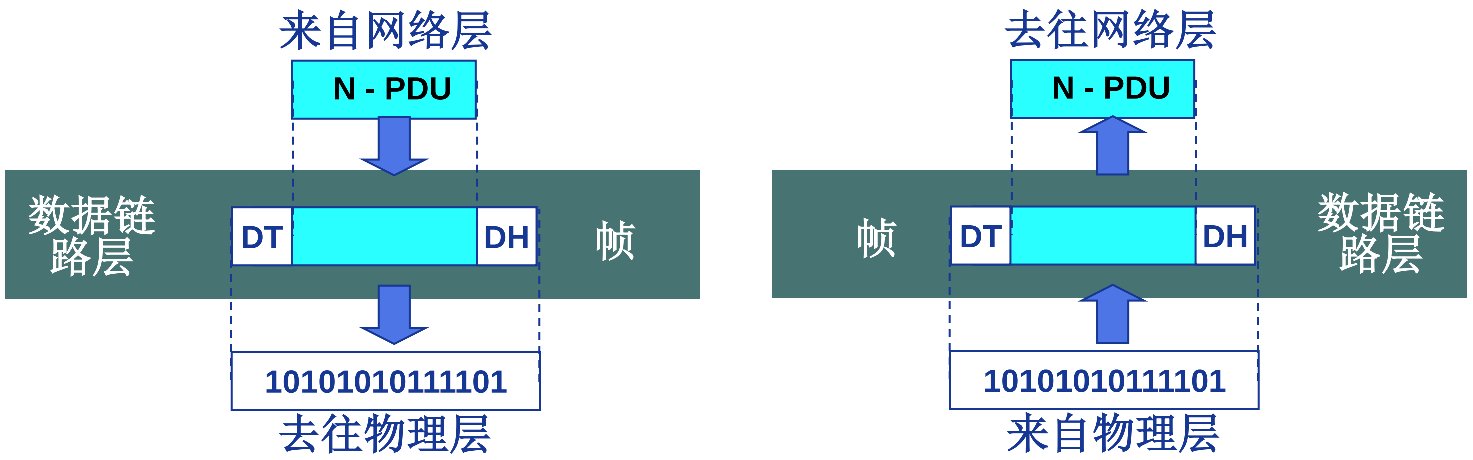
为什么需要数据封装？

- 增加控制信息
  - 构造协议数据单元 (PDU)
- 控制信息主要包括:
  - 地址（Address）: 标识发送端/接收端
  - 差错检测编码（Error-detecting code）： 用于差错检测或纠正
  - 协议控制（Protocol control）: 实现协议功能的附加信息，如: 优先级（priority）、服务质量（QoS）、 和安全控制等

物理层功能

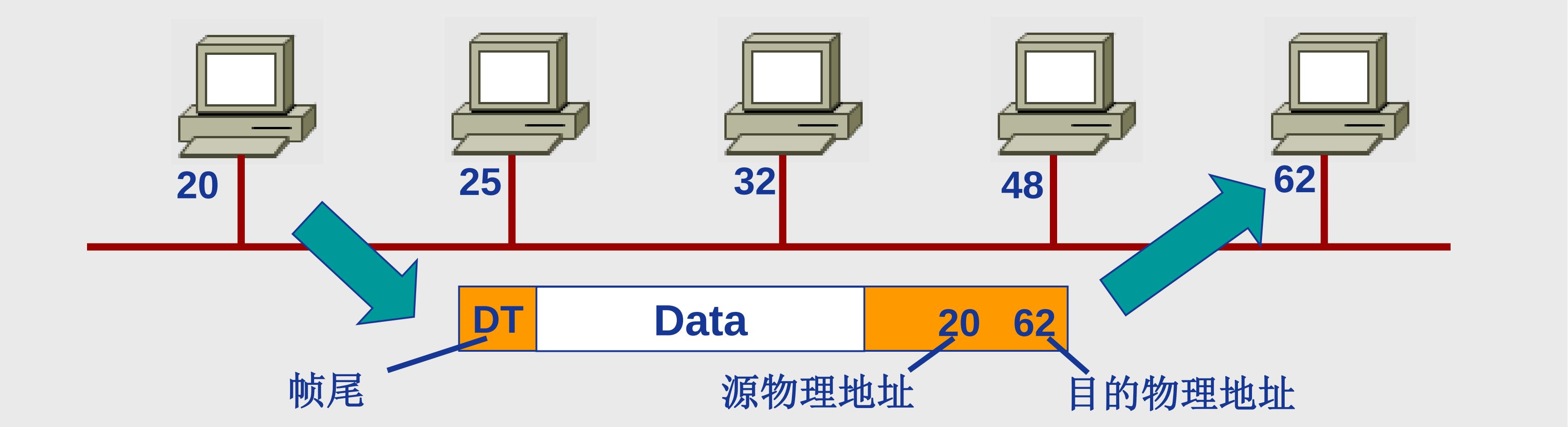
- 接口特性
  - 机械特性、电气特性、功能特性、规程特性
- 比特编码
- 数据率
- 比特同步
  - 时钟同步
- 传输模式
  - 单工（Simplex）
  - 半双工（half-duplex）
  - 全双工（full-duplex）

数据链路层功能



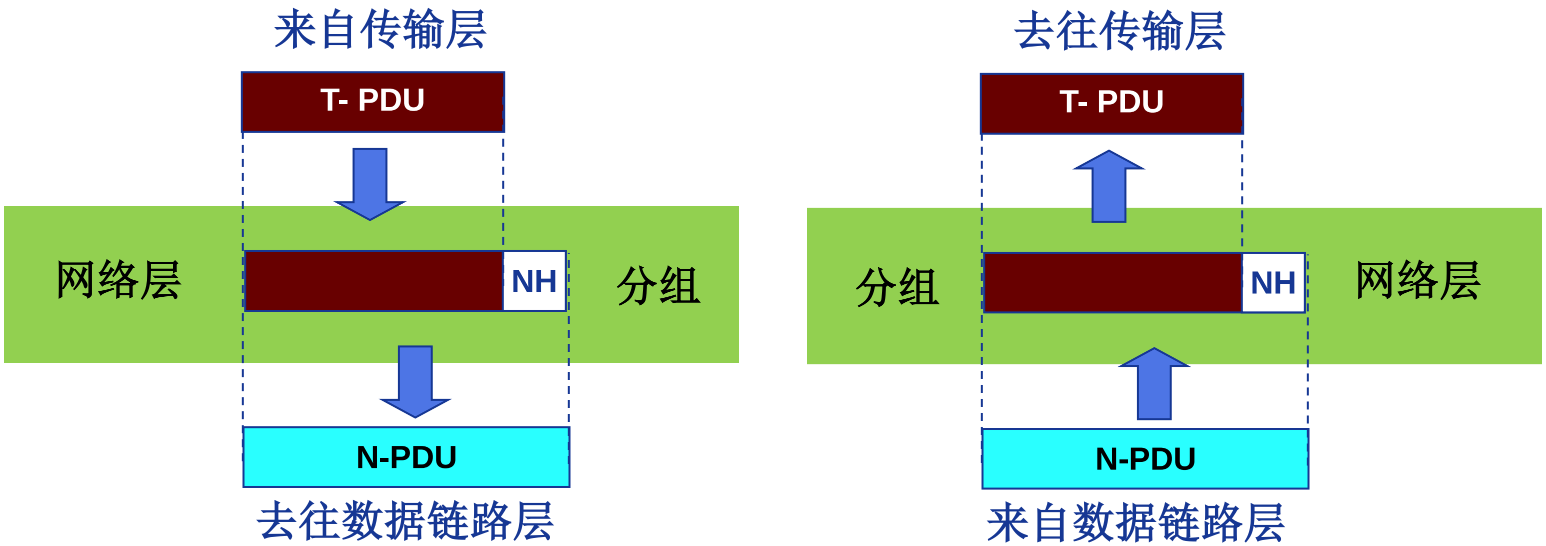
- 负责结点-结点（node-to-node）数据传输
- 组帧（Framing）
- 物理寻址（Physical addressing）
  - 在帧头中增加发送端和/或接收端的物理地址标识数据帧的发送端和/或接收端

数据链路层功能



- 流量控制（Flow control）
  - 避免淹没接收端
- 差错控制（Error control）
  - 检测并重传损坏或丢失帧，并避免重复帧
- 访问控制（Access control）
  - 在任一给定时刻决定哪个设备拥有链路（物理介质）控制使用权

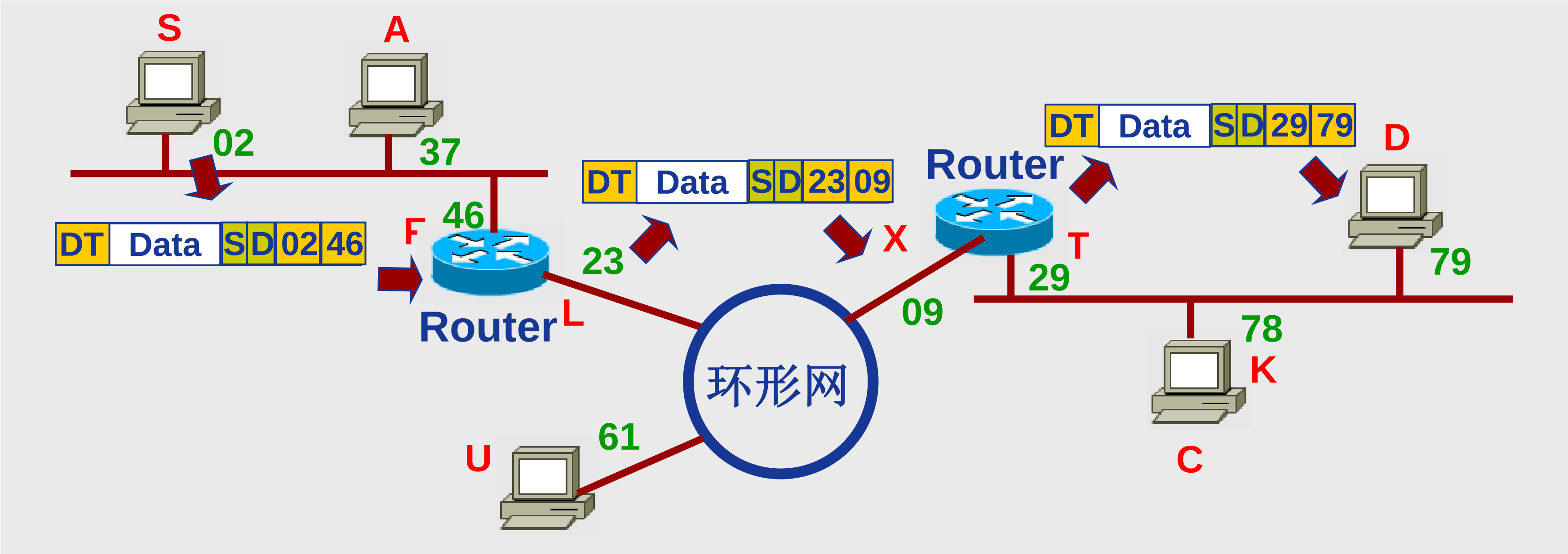
网络层功能



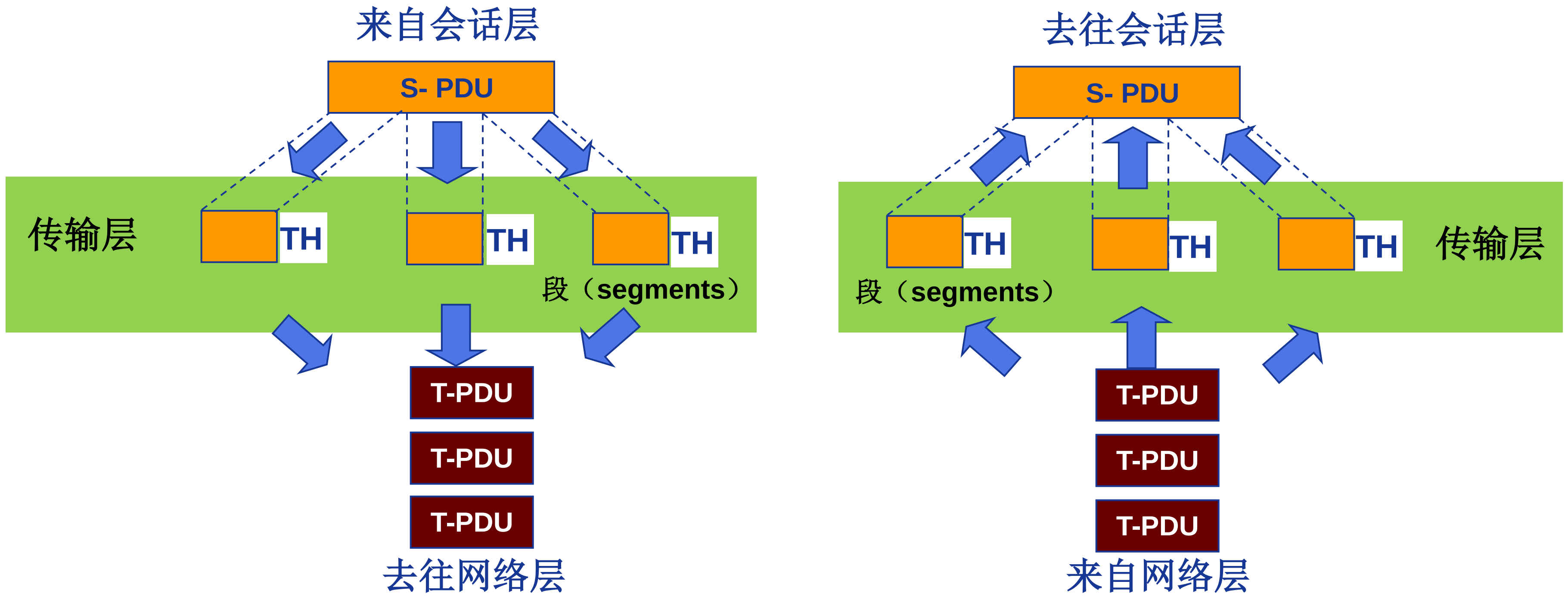
- 负责源主机到目的主机数据分组（packet）传输
  - 可能穿越多个网络
- 逻辑寻址（Logical addressing）
  - 全局唯一逻辑地址，确保数据分组被送达目的主机，如IP地址

网络层功能

- 路由（Routing）
  - 路由器（或网关）互连网络，并路由分组至最终目的主机

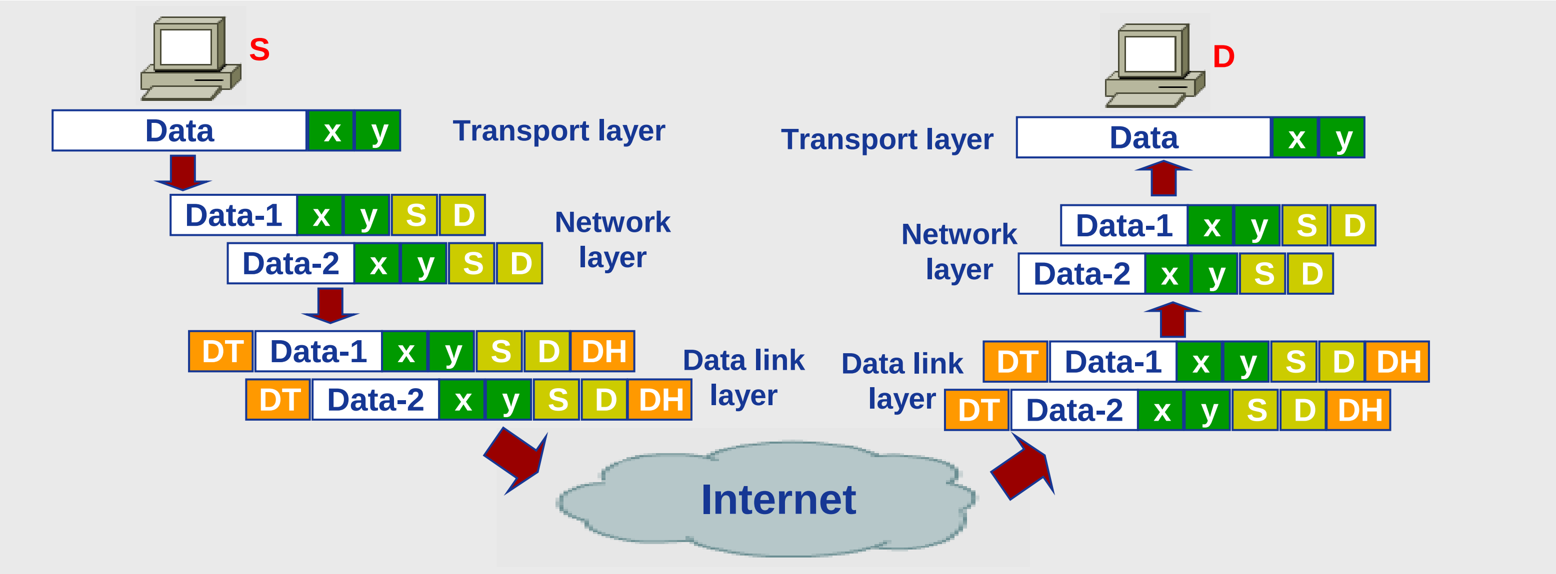


传输层功能



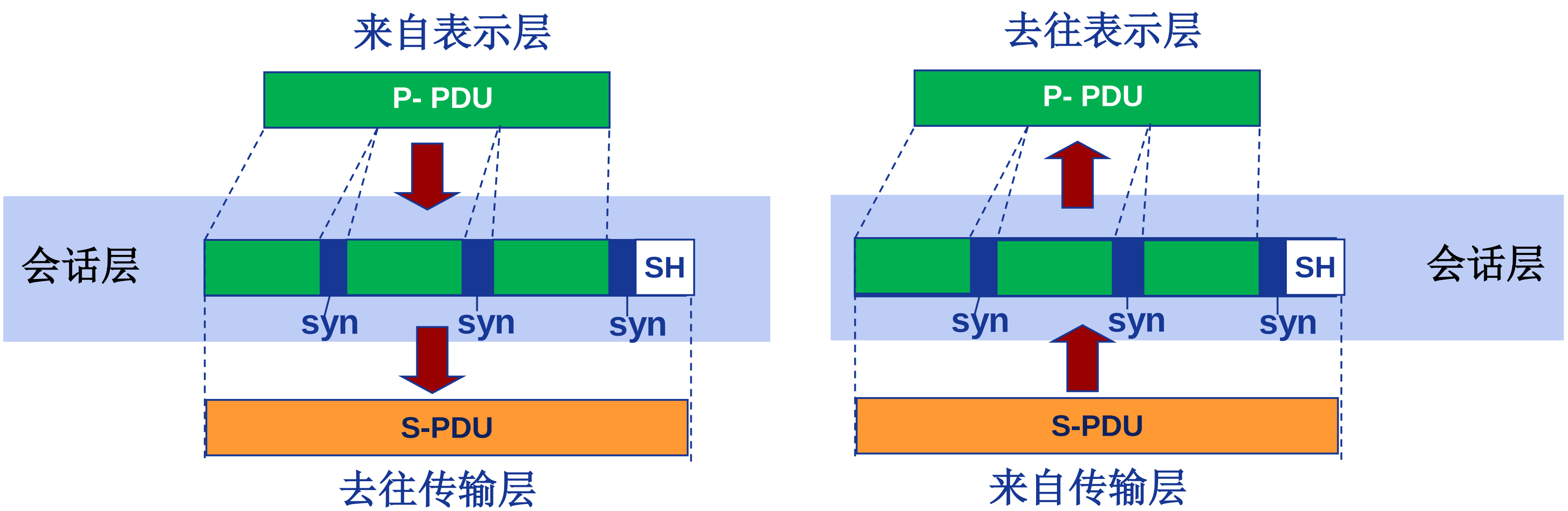
负责源-目的（端-端）（进程间）完整报文传输

传输层功能



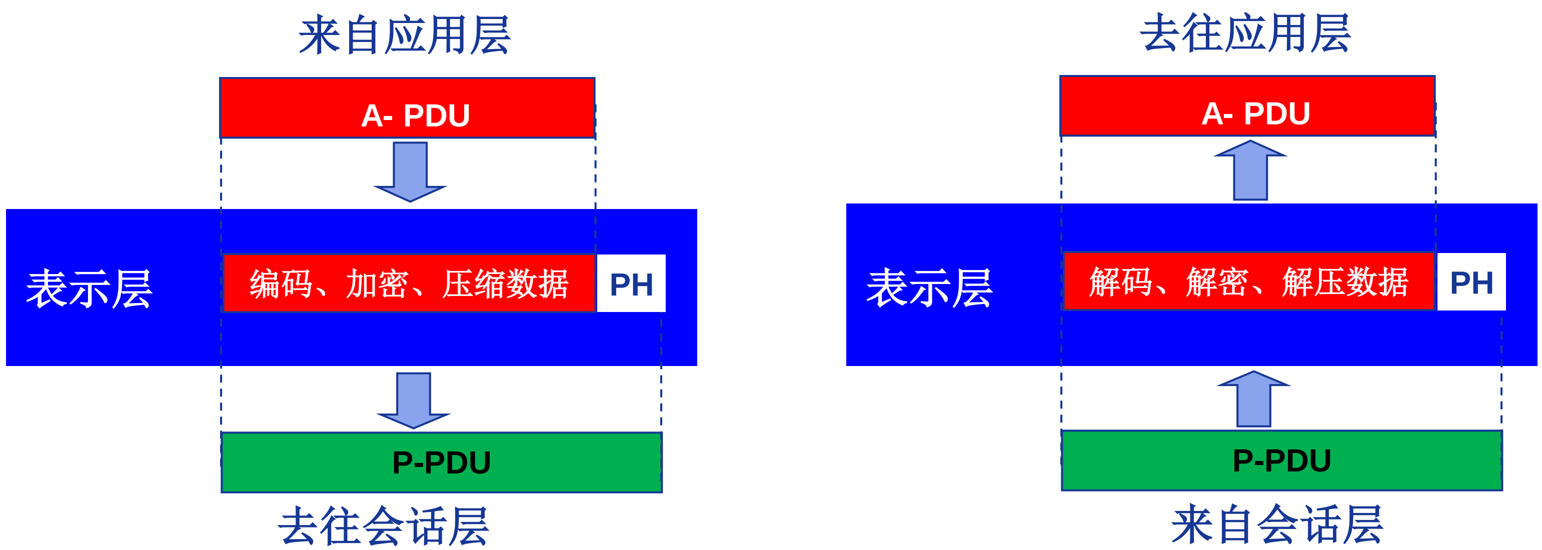
- 分段与重组
- SAP寻址
  - 确保将完整报文提交给正确进程，如端口号
- 连接控制
- 流量控制
- 差错控制

会话层功能



- 对话控制（dialog controlling）
  - 建立、维护
- 同步
  - 在数据流中插入“同步点”
- 最“薄”的一层

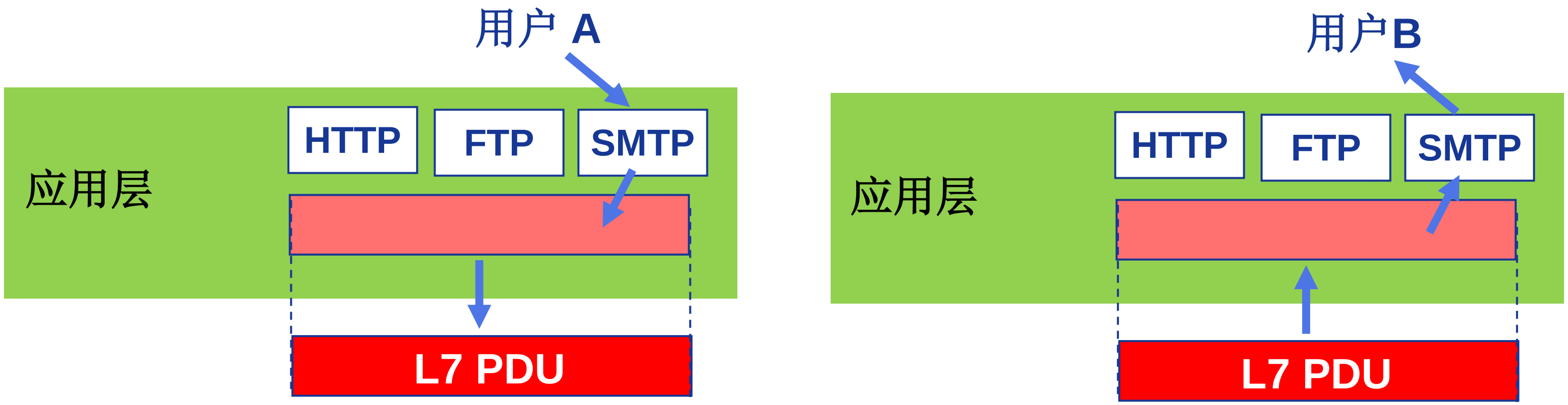
表示层功能



处理两个系统间交换信息的**语法与语义（syntax and semantics ）**问题

- 数据表示转化
  - 转换为主机独立的编码
- 加密/解密
- 压缩/解压缩

应用层功能

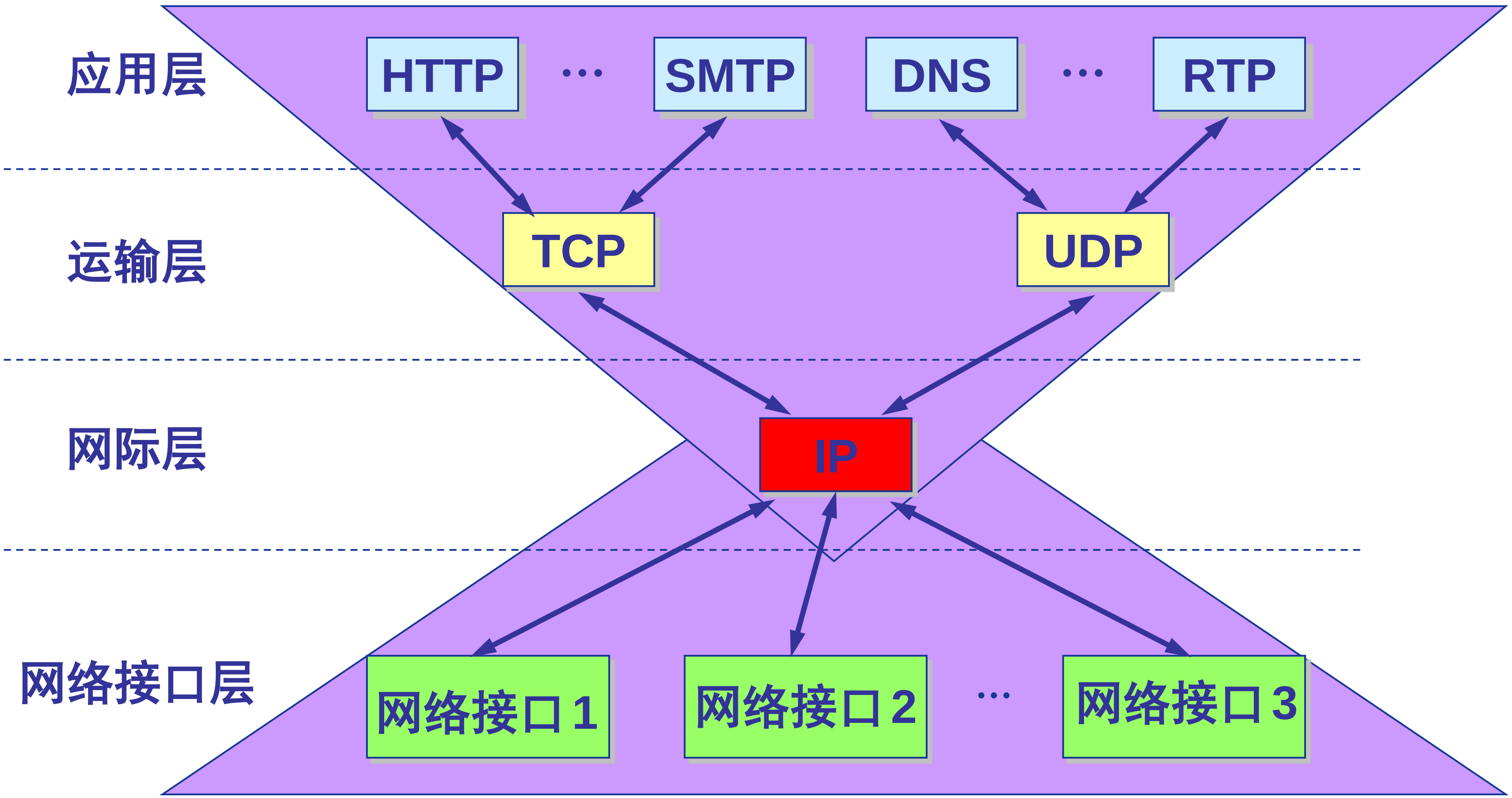


- 支持用户通过用户代理（如浏览器）或网络接口使用网络（服务）
- 典型应用层服务：
  - 文件传输（FTP）
  - 电子邮件（SMTP）
  - Web（HTTP）
  - ……

TCP/IP参考模型

IP over Everything

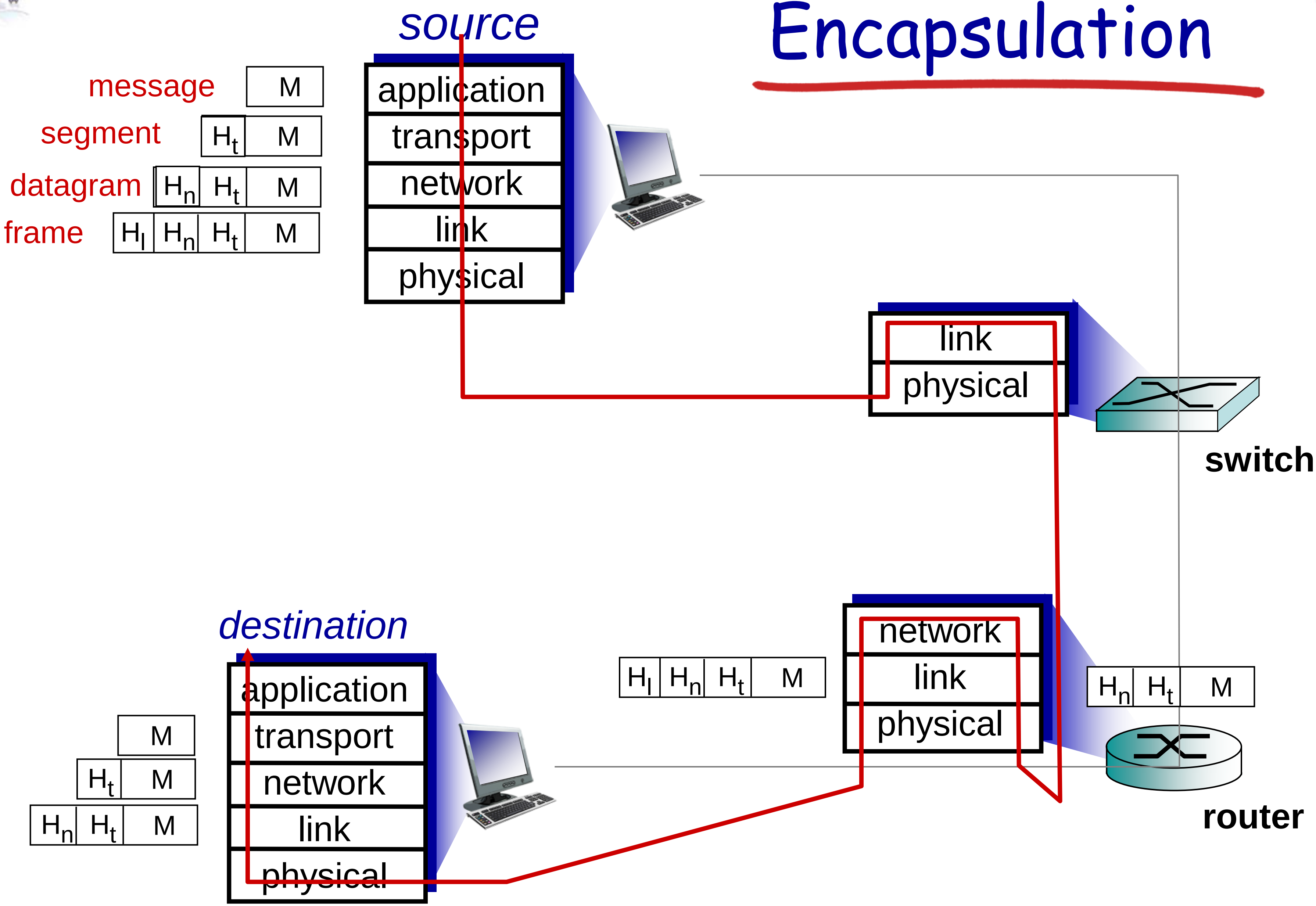
IP 可应用到各式各样的网络上



5层参考模型

- 综合 OSI 和 TCP/IP 的优点
- *应用层*: 支持各种网络应用
  - FTP, SMTP, HTTP
- *传输层*: 进程-进程的数据传输
  - TCP, UDP
- *网络层*: 源主机到目的主机的数据分组路由与转发
  - IP协议、路由协议等
- *链路层*: 相邻网络元素（主机、交换机、路由器等）的数据传输
  - 以太网（Ethernet）、802.11 (WiFi)、PPP
- *物理层*:比特传输

Encapsulation



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

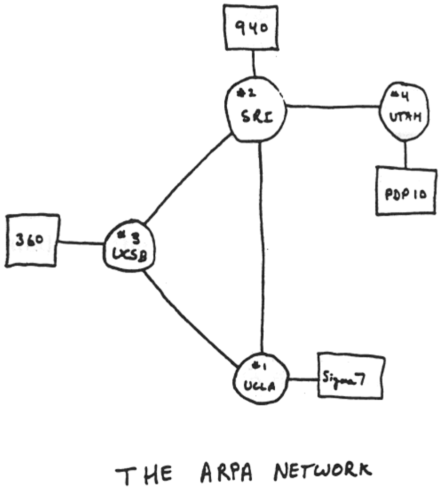
1.5 protocol layers, service models

1.6 history

Internet history

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational
- 1972:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet history

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control

- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet history

1990, 2000's: commercialization, the Web, new apps

- early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

late 1990's – 2000's:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Internet history

2005-present

- ~750 million hosts
  - Smartphones and tablets
- Aggressive deployment of broadband access
- Increasing ubiquity of high-speed wireless access
- Emergence of online social networks:
  - Facebook: soon one billion users
- Service providers (Google, Microsoft) create their own networks
  - Bypass Internet, providing "instantaneous" access to search, emai, etc.
- E-commerce, universities, enterprises running their services in "cloud" (eg, Amazon EC2)

Introduction: summary

covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- history

you now have:

- context, overview, "feel" of networking
- more depth, detail *to follow!*